



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

***“Cálculo de aerogenerador en celosía mediante
software basado en Visual Basic”***

MEMORIA

Javier Vergara Lecue

Tutores: Cesar Díaz de Cerio

Pamplona, 24 de Noviembre de 2010

Índice

Página

PARTE 1: INTRODUCCIÓN Y OBJETIVOS..... 8

1. INTRODUCCIÓN	9
2. OBJETIVO DEL PROYECTO.....	11
3. ESQUEMA DE LOS CÁLCULOS	12
3.1 PLANTEAMIENTO GENERAL	12
3.2 DESCRIPCIÓN DE CADA RUTINA DE CÁLCULOS	12
4. ESQUEMA DEL PROGRAMA	12

PARTE 2: CONCEPTOS TEÓRICOS..... 14

1. ESQUEMA GENERAL DE LOS CÁLCULOS DEL PROGRAMA	15
2. INTRODUCCIÓN DE DATOS	16
2.1 Datos torre.....	16
2.2 Datos de cimentación	18
2.3 Datos Tramos de Torre	18
3. DIVISIÓN DE LA TORRE EN INCREMENTOS	22
3.1 División torre	22
3.2 Altura incrementos	22

3

3.3 Suma de pesos.....	23
3.4 Explicación rutina	24
4. CÁLCULO DE LOS COEFICIENTES DINÁMICOS Y DE EMPUJE DEL VIENTO	25
4.1 Coeficiente de fuerza	25
4.1 Coeficiente dinámicos.....	28
4.3 Explicación rutina	34
5. CÁLCULO DE LA FUERZA DE VIENTO EJERCIDA EN CADA INCREMENTO	41
5.1 Cálculo de la fuerza.....	41
5.2 Explicación rutina	42
6. CÁLCULO DE LOS MOMENTOS FLECTORES, FUERZAS CORTANTES Y PESOS	47
6.1 Cálculo de momentos debidos al viento en la góndola.....	48
6.2 Cálculo de momentos debidos al viento en las aspas.....	49
6.3 Cálculo de momentos debidos al peso descentrado de góndola y aspas.....	50
6.4 Cálculo de momentos debidos al viento en tramos de torre	52
6.5 Fuerzas cortantes debidas al viento.....	54
6.6 Explicación rutina	55
7. CÁLCULO DE LAS FLECHAS DE DEFORMACIÓN CAUSADOS POR LAS DISTINTAS SOLICITACIONES	58
7.1 Cálculo del momento de inercia de la sección respecto al eje transversal y diagonal.....	60
7.2 Cálculo de las flechas.....	63
7.3 Explicación rutina	66

8. CÁLCULO DE LOS MOMENTOS SECUNDARIOS CAUSADOS POR LAS DISTINTAS SOLICITACIONES	68
8.1 Explicación rutina	69
9. CÁLCULO DE ESFUERZOS EN CADA TRAMO DE TORRE, MONTANTES Y DIAGONALES.....	71
9.1 Flexión pura en dirección transversal a la sección. Caso 1.....	71
9.2 Flexión pura en dirección diagonal a la sección. Caso 2	75
9.3 Momentos flectores con fuerzas cortantes en dirección transversal a la sección. Caso 1.....	79
9.3.1 Esfuerzos en tramo Tipo 1	80
9.3.2 Esfuerzos en tramo Tipo 2	88
9.4 Momentos flectores con fuerzas cortantes en dirección transversal a la sección. Caso 2.....	97
9.4.1 Esfuerzos en tramo Tipo 1	97
9.4.2 Esfuerzos en tramo Tipo 2	107
9.5 Resumen esfuerzos.....	116
9.6 Explicación Rutina.....	122
10. CÁLCULOS VARIOS EN LA BASE.....	126
10.1 Caso 1.....	128
10.2 Caso 2.....	129
10.1 Explicación rutina	131
11. CÁLCULOS CIMENTACIÓN	109
11.1 Comprobación al vuelco y al deslizamiento	109
11.2 Distribución de presiones del terreno.....	110
11.3 Cálculos.....	111

11.4 Explicación rutina	115
12. SALIDA DE RESULTADOS	116

PARTE 1

INTRODUCCIÓN Y OBJETIVOS

1 INTRODUCCIÓN

Se conoce como energía eólica al aprovechamiento por el hombre del viento transformándolo en energía más utilizable para él. Hoy en día se emplea sobre toda para generar energía limpia y segura. Es una energía limpia, lo que quiere decir que la generación de electricidad a partir del viento no produce gases tóxicos, no origina productos secundarios peligrosos, ni residuos contaminantes.

En la actualidad existe toda una enorme variedad de modelos de aerogeneradores, diferentes entre sí tanto por la potencia proporcionada, como por el número de palas o incluso por la manera de producir energía eléctrica (aisladamente o en conexión directa con la red de distribución convencional).

Los aerogeneradores son siempre elementos muy visibles en el paisaje. De lo contrario, no estarían situados adecuadamente desde un punto de vista meteorológico.

En todo caso el impacto visual es una consecuencia directa de esta forma de producir energía. Puede minimizarse en lo posible, por ejemplo pintando las torres de gris, pero nunca evitarse totalmente. En áreas llanas suele ser una buena estrategia disponer las turbinas según una distribución geométrica simple, fácilmente perceptible por el espectador.

Sin embargo, existen límites a la utilización de patrones simples: en paisajes con fuertes pendientes, rara vez es viable la utilización de una distribución geométrica simple, y suele ser mejor hacer que las turbinas sigan los contornos del altitud del paisaje, o los cercados u otras características del mismo.

Otro efecto achacado a este tipo de instalaciones es la contaminación acústica. Aunque el sonido no es un problema capital para la industria, dada la distancia a la que se encuentran los vecinos más cercanos, no por ello es este un detalle que se descuide totalmente a la hora de diseñar nuevos equipos. Además, ningún paisaje está nunca en silencio absoluto. Por ejemplo, las aves y las actividades humanas emiten sonidos y, a velocidades del viento alrededor de 4-7 m/s y superiores, el ruido del viento en las hojas, arbustos, árboles, mástiles, etc. ahogará gradualmente cualquier potencial sonido de los aerogeneradores.

Por último se hace necesario reseñar, un debate abierto entre la industria explotadora de parque eólicos y los diferentes grupos de defensa de la naturaleza, quienes paradójicamente deberían ser los mayores defensores de esta fuente no contaminante de energía. Tal debate es la mortandad de aves causadas por colisiones con las aspas de los equipos. Las posturas son enfrentadas y las posiciones dispares.

La energía eólica es una forma indirecta de energía solar, puesto que son las diferencias de temperatura y de presión inducidas en la atmósfera por la absorción de la radiación solar las que ponen en movimiento los vientos. Se calcula que un 2 % de la energía solar recibida por la Tierra se convierte en energía cinéticas de los vientos. La cantidad de energía correspondiente es enorme: unos 30 millones de TWh por año. Incluso teniendo en cuenta que sólo el 10 % de esta energía se encuentra disponible cerca del suelo, el potencial sigue siendo considerable; así, es difícil concebir en la actualidad la explotación de una parte notable de este potencial. En efecto, sería necesario cubrir las tierras emergidas y las superficies marinas con enormes motores eólicos. En estas condiciones, es más razonable estimar que por mucho tiempo las aplicaciones de la energía eólica se limitaran a utilizaciones locales, en regiones aisladas - a un nivel de potencia de algunos Kw. a algunas decenas de Kw.- o bien a un papel de fuente complementaria en la alimentación de las redes eléctricas - con niveles de potencia de hasta algunos MW-. Las zonas más favorables para la implantación de grandes motores eólicos son las regiones costeras y las grandes estepas, donde vientos constates soplan regularmente: es necesaria una velocidad media del viento superior a 30 km/h.

La torre del aerogenerador soporta la góndola y el rotor.

La mayoría de los grandes aerogeneradores se entregan con torres tubulares de acero, fabricadas en secciones de 20-30 metros con bridas en cada uno de los extremos, y son unidos con pernos "in situ". Las torres son tronco-cónicas (es decir, con un diámetro creciente hacia la base), con el fin de aumentar su resistencia y al mismo tiempo ahorrar material. Pero este tipo de torres tienen un gran coste de construcción y sobre todo de transporte. Por ello se está planteando poco a poco la posibilidad de hacer torres en celosía. Las torres de celosía son fabricadas utilizando perfiles de acero soldados. La ventaja básica de las torres de celosía es su coste, puesto que una torre de celosía requiere sólo la mitad de material que una torre tubular sin sustentación adicional con la misma rigidez. La principal desventaja de este tipo de torres es su apariencia visual (aunque esa cuestión es claramente debatible).

2 OBJETO DEL PROYECTO

Este proyecto está realizado para realizar una serie de cálculos del aerogenerador con una torre en celosía. Los datos debidos referidos a la góndola y las palas del rotor, se introducen en el programa como datos conocidos, como suele ser la fuerza del viento, el peso...

Una vez centrados en el estudio de la torre, podemos decir que está formada por un conjunto de tramos de torre que se van acoplando uno encima de otro hasta alcanzar la altura y prestaciones exigidas.

El tipo de tramo de torre elegido es función de las condiciones a las que estará sometido el aerogenerador, debido sobre todo a la zona geográfica en donde se instalará. No es lo mismo instalar un parque eólico junto al mar, que en la cumbre de una montaña o en una zona árida. Todos estos lugares geográficos tienen distintos efectos sobre el aerogenerador. De manera que el conjunto de la torre puede ser distinto para cada uno de estos lugares.

Así pues no es nuestra función el diseño de tipos de tramos, sino que debemos de elegir lo más adecuados de entre los ya existentes para obtener unas prestaciones máximas con el mínimo coste, además de asegurarnos de que existe un margen de seguridad adecuado.

Este listado de tramos de torre se tomará de una base de datos externa. Lo que se ha hecho es crear una serie de tramos de torre, con una serie de propiedades como altura, sección... que se describen más adelante. Cualquier usuario del programa podrá crear su propio tramo de torre, siempre que se ajuste a la geometría que se ha especificado en el programa.

El fin de este proyecto es crear la base de un programa de cálculo rápido y sencillo, en el cual en el futuro fácilmente se puede adaptar fácilmente a numerosos cambios sin que ello cambie la estructura del programa. Como se ha comentado anteriormente no tiene como fin el diseño de tipos de tramos, si no que el estudio de cálculo de ellos.

3 ESQUEMA DE LOS CÁLCULOS

Mediante los cálculos se describen los conceptos que soportan la elaboración del programa que constituye el proyecto. En los siguientes pasos se encuentran las explicaciones oportunas.

3.1 PLANTEAMIENTO GENERAL

En un principio se exponen los pasos fundamentales para poder obtener los resultados deseados. Se intenta dar una pequeña idea de lo que es el programa en sí, sin profundizar en los métodos de cálculo, es decir un pequeño esquema, con el fin de comprender el proceso y así saber en cada momento en qué situación se encuentra el cálculo frente al conjunto global. Si no se corre el riesgo de que si explican todos los pasos de forma consecutiva podría dar lugar a una desorientación y pérdida de la visión de conjunto.

3.2 DESCRIPCIÓN DE CADA RUTINA DE CÁLCULOS

En este apartado es donde se detalla todo el proceso y cálculos elegidos, tanto las razones de su elección, declaración de variables... Se intenta profundizar en todo momento para que no haya ninguna duda y queden todos los procesos claros. Para ello se escriben los algoritmos o rutinas de cálculo y se explican paso a paso.

4 ESQUEMA DEL PROGRAMA

La función principal del programa es la de realizar los cálculos oportunos. Pero como todo programa que se precie hoy en día, debe de tener opciones tales como Abrir, Guardar, Guardar Como... y que visualmente sea agradable y fácil de utilizar. Estas opciones también se explican, pero en otro apartado, fuera del planteamiento general de cálculo. También se detalla las propiedades de todos los elementos que forman el programa.

PARTE 2:

CONCEPTOS

TEÓRICOS

1 ESQUEMA GENERAL DE LOS CÁLCULOS DEL PROGRAMA

Exponemos los pasos fundamentales de los cálculos, y para ello con el siguiente esquema se intenta dar una pequeña idea de lo que es el programa de cálculo en sí. Así en todo momento se puede saber en qué punto de los cálculos nos encontramos sin llegar a confusiones, y teniendo siempre una visión del conjunto.

El proceso seguido es el siguiente:

- Introducción de datos.
- División de la torre en incrementos
- Cálculo de los coeficientes dinámico y empuje de viento
- Cálculo de la fuerza de viento ejercida en cada incremento
- Cálculo de los momentos flectores, fuerzas cortantes y pesos
- Cálculo de las flechas de deformación causadas por las distintas solicitaciones
- Cálculo de los momentos secundarios causados por las distintas solicitaciones
- Cálculos esfuerzos en cada tramo de torre, en montantes y diagonales.
- Cálculos varios en la base
- Cálculos de cimentación
- Salida de resultados

2 INTRODUCCIÓN DE DATOS

Se explican todos los datos que se deben introducir en el programa para poder realizar todos los cálculos necesarios.

Tendremos tres grupos. Datos de la torre en general, datos sobre los tramos de celosía y datos sobre la cimentación.

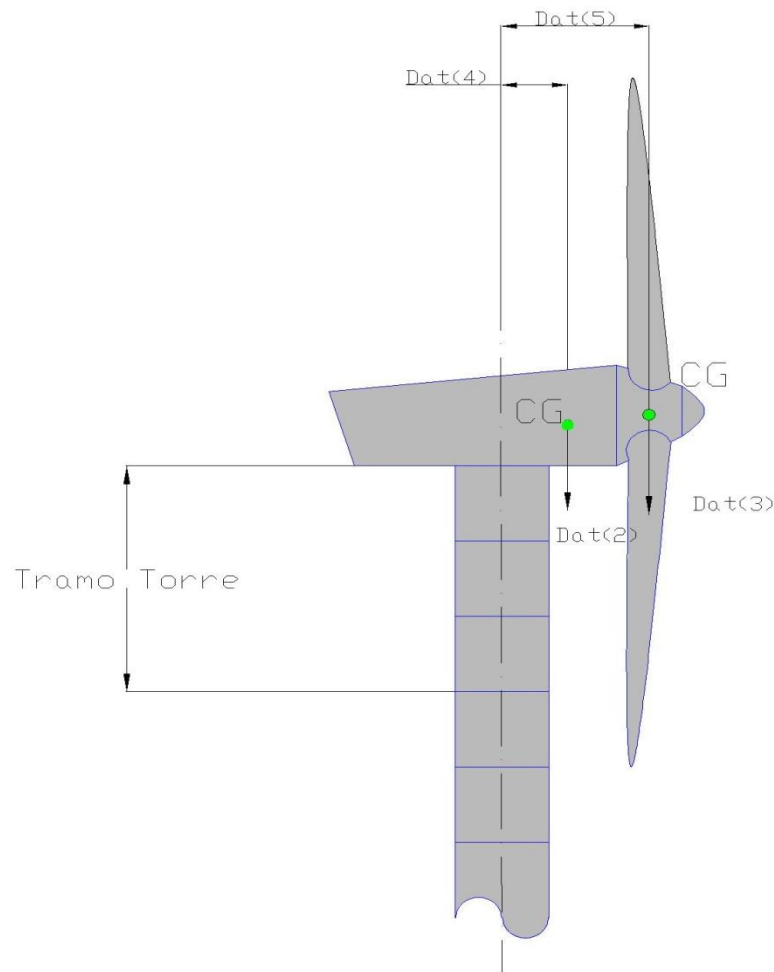
Los datos de la torre en general y los datos sobre la cimentación son en total 13 y los guardamos en la variable Dat(13).

2.1 DATOS DE TORRE

- Dat(1): Número de tramos de torre. Este dato no puede ser introducido manualmente. Depende del número de tramos que vayamos introduciendo mediante la lista desplegable o conociendo el código del tramo. Cada vez que insertemos un tramo de torre se suma la unidad, y si eliminamos algún tramo se resta la unidad.
- Dat(2): Peso Góndola: Para poder calcular el momento que genera el peso de la góndola debido a su excentricidad respecto a la torre, debemos de conocer su peso. Además es necesario para poder calcular las reacciones en la torre debido a las distintas solicitaciones.
- Dat(3): Peso Aspas: Ocurre lo mismo que con el peso de la góndola. Debemos de conocer el peso de las aspas para conocer el momento que genera el peso descentrado de las aspas respecto a la torre y también para calcular correctamente las reacciones en la torre.
- Dat(4): Distancia CG de la Góndola respecto al eje de la torre. Debemos de saber esta distancia para conocer la distancia de excentricidad que existe entre el centro de la torre y el centro de gravedad de la góndola.
- Dat(5): Distancia hasta aspas: Suponemos que esta distancia es desde el centro de eje de la torre hasta el eje de rotor de las aspas. Para hacer un cálculo exacto deberíamos de saber el centro de gravedad de las aspas. Pero si damos esta distancia, damos una distancia mayor de lo que es en realidad, con lo que al hacer los cálculos tenemos un margen mayor de seguridad.
- Dat(6): Superficie Frontal de Góndola: Este dato se utiliza para comprobar la fuerza que el viento ejerce sobre la Góndola. Para ello es necesario saber sobre que superficie incide el viento.

- Dat(7): Altura de la Góndola: Este dato también se utiliza para comprobar la fuerza que el viento ejerce sobre la góndola. Cuanto más alto está un elemento expuesto al viento mayor es la velocidad que tiene el viento, con lo cual también es mayor la fuerza del viento. Suponemos que la fuerza del viento esta ejercida justamente en mitad de la altura de la góndola. Por ello necesitamos este dato.
- Dat(8): Fuerza Viento Sobre aspas: Dada la complejidad de la forma de las aspas, este dato se introduce manualmente, sin que el programa tenga que calcularlo.
- Dat(9): Viento de Referencia: Según en la zona geográfica en que nos encontremos, tenemos distintos vientos de referencia, por lo que este dato se introduce manualmente, para comprobar cómo reacciona una misma torre en distintos lugares.
- Dat(10): Coeficiente peso propio:

La introducción de algún dato puede que quede más claro con la siguiente figura



2.2 DATOS DE CIMENTACIÓN

- Dat(11): Densidad de Hormigón: Este dato suele ser de 2400 Kg/m^3 . Pero puede que se utilice otro material u otro tipo de hormigón, con lo que se da la opción de introducirlo.
- Dat(12): Tensión admisible del terreno.
- Dat(13): Altura zapata: Para poder dimensionar la zapata correctamente, debemos de introducir que altura tiene.

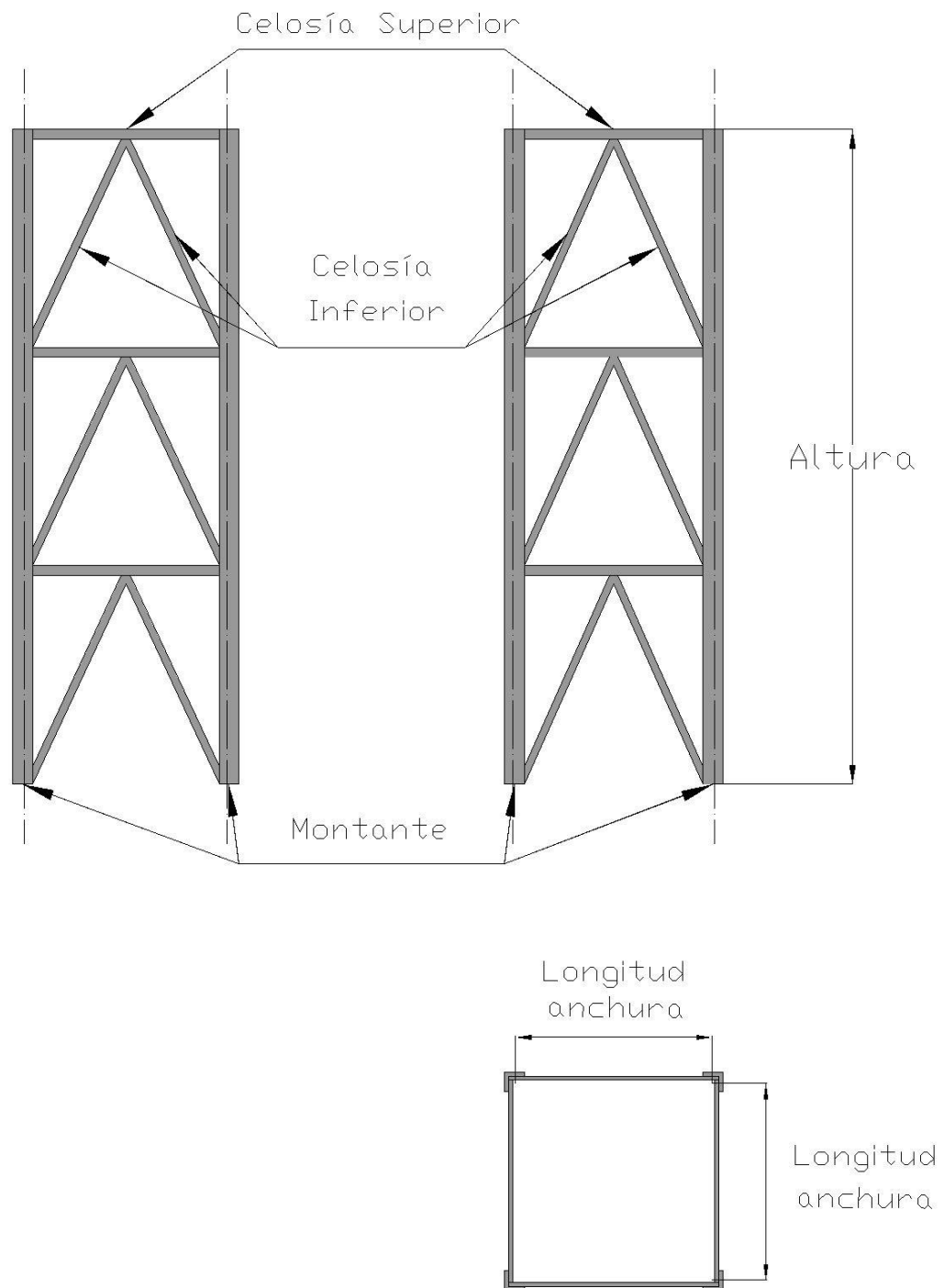
2.3 DATOS TRAMOS DE TORRE

Los datos que se especifican en la base de datos externa y por lo tanto que utiliza el programa son los siguientes:

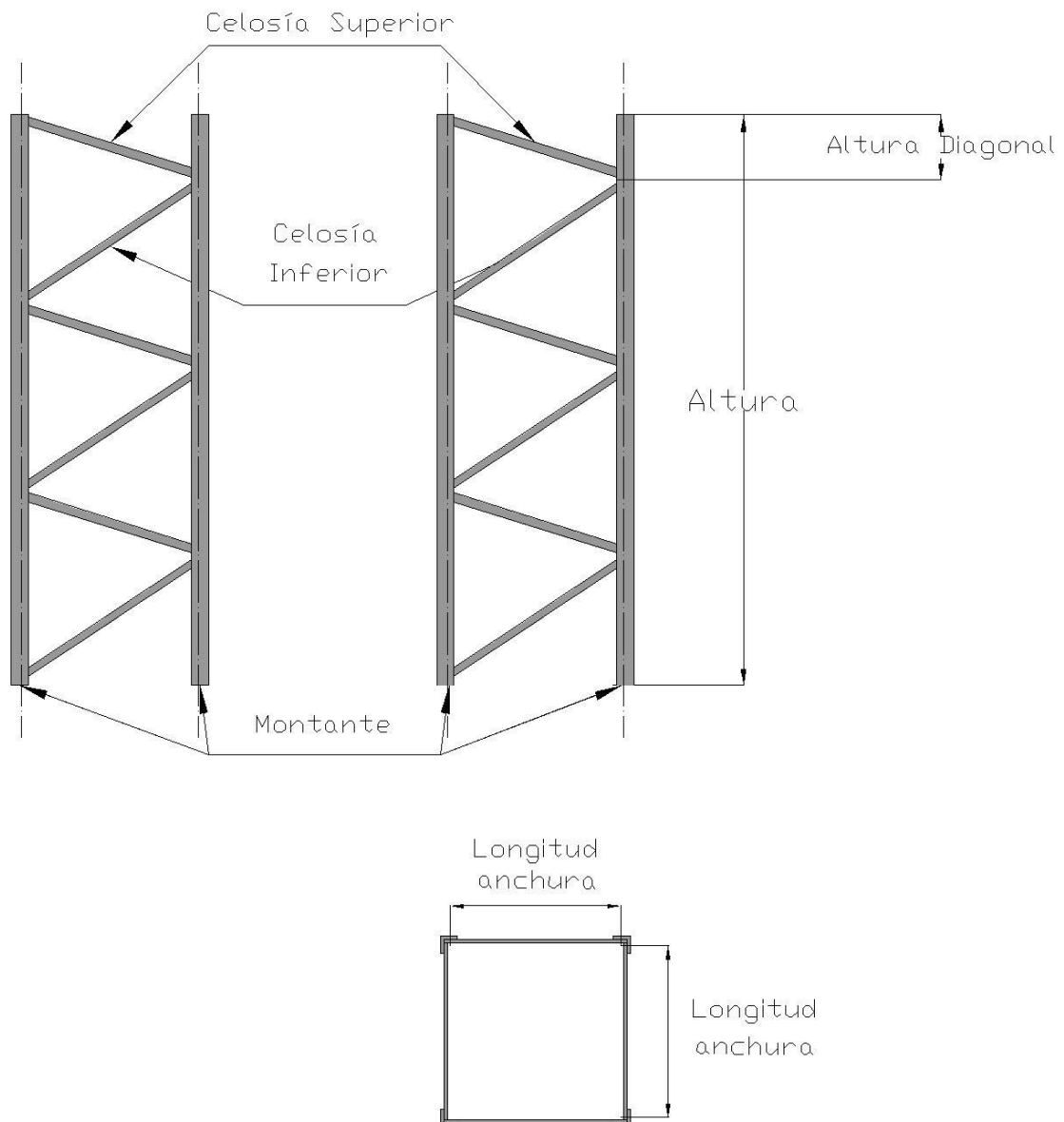
- Código: Nombre de identificación del tramo.
- Número de pasos: Cada tramo se compone de una serie de pasos. Y estos por su parte se compone de una celosía superior, una inferior y la parte de montante correspondiente a dichas diagonales.
- Altura: Longitud total del tramo en m.
- Longitud de anchura: Distancia entre los centros de gravedad de cada montante. Es una distancia muy parecida al lado del montante. Esta distancia sirve para calcular los momentos de inercia de cada tramo sin tener en cuenta las diagonales.
- Peso: Peso del tramo en Kg.
- Sección del montante en cm^2 .
- Sección de la celosía superior cm^2 .
- Sección de la celosía inferior cm^2 .
- Área superficie del viento en el tramo m^2 : Es la superficie frontal del tramo que queda expuesta al viento.
- Tipo de tramo.
- Altura diagonal. Sólo para el tramo Tipo 2.

Todos estos datos quedan más claros al ver la siguiente figura.

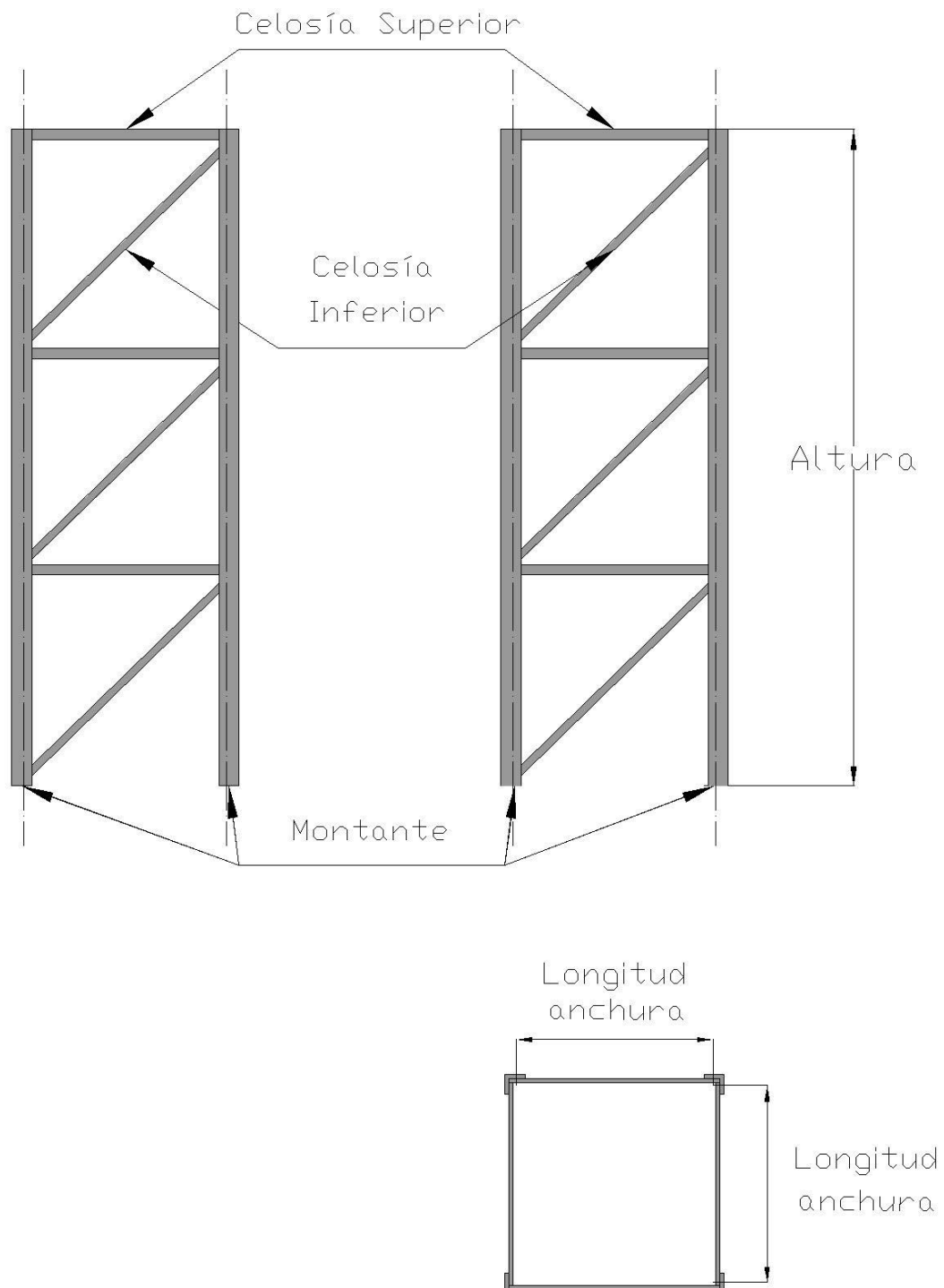
TRAMO TIPO 1



TRAMO TIPO 2



Si en el tramo Tipo 2 le damos un valor de 0 a la altura de la diagonal obtenemos un tramo del tipo:



3 DIVISIÓN DE LA TORRE EN INCREMENTOS

A lo largo de toda la torre, al hacer los cálculos, asignamos valores tales como momentos, fuerzas ejercidas por el viento, flechas... Para ello dividimos la torre en un número de incrementos. Se ha optado por dividir la torre en 100 incrementos, pensado que la exactitud, rapidez de cálculo y cantidad de memoria utilizada por el ordenador será aceptable.

3.1 NÚMERO INCREMENTOS

Al hacer la división, asignamos a cada tramo de torre el mismo número de incrementos. En consecuencia como la altura de los tramos de torre puede variar, las alturas de los incrementos serán distintas de un tramo de torre a otro.

El número de incrementos por tramo queda almacenado en la variable $NInc(1)$ y es igual a:

$$NInc(1) = Int\left(\frac{100}{Dat(1)}\right)$$

- Nota: Int . Función del programa Visual Basic que devuelve la parte entera

El número de incrementos totales utilizados por el programa quedará almacenado en la variable $NInc(2)$ y es igual a:

$$NInc(2) = (Dat(1)) \cdot NInc(1)$$

3.2 DIVISIÓN ALTURA TORRE

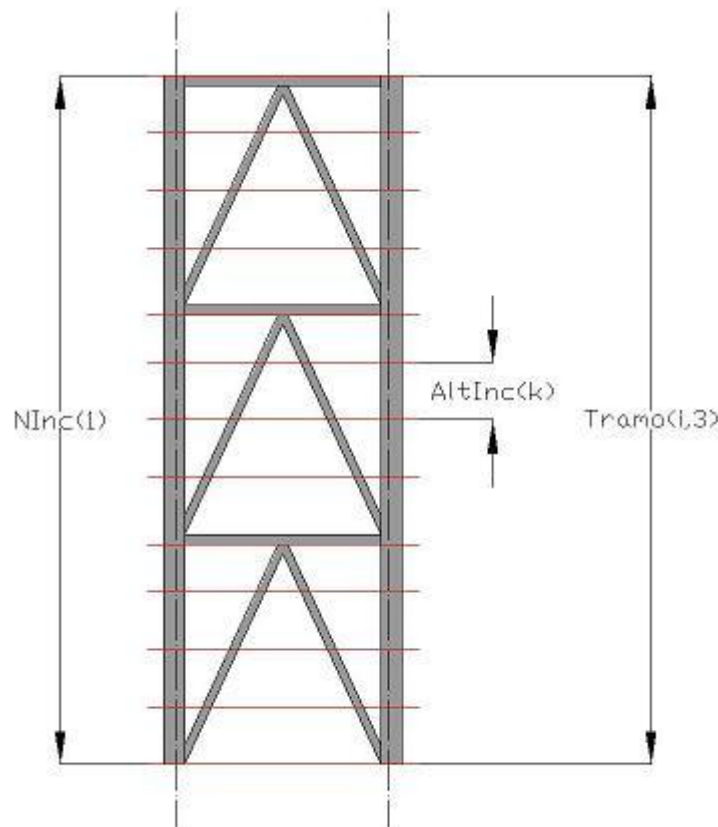
Como he comentado antes la altura de estos incrementos varía a la largo de la torre, de un tramo de torre a otro. De manera que debemos de calcular estas alturas y guardarlas en la variable $AltInc(k)$. k es el número de incremento, de manera que la variable $AltInc(k)$ es un vector con 100 espacios de memoria.

Hay una cosa que debemos de tener en cuenta. Todos los cálculos que realizamos empiezan por la parte superior de la torre. De manera que el primer incremento de la torre es aquel que está en la punta, y el último incremento está en la base.

La altura de un incremento k dentro de un tramo i es igual a la altura del tramo dividido por el número de incrementos por tramo. Esto lo conseguiremos con dos bucles *For Next*

$$AltInc(k) = \frac{Tramo(i, 3)}{NInc(1)}$$

$Tramo(i, 3) = \text{Altura del Tramo } i$



3.3 SUMA DE PESOS.

En la variable $Peso(100)$ almacenamos los pesos por encima del incremento que estamos calculando. Para un incremento k , el valor del peso se almacena en la variable $Peso(k)$.

El peso del incremento se consigue sumándole el valor del peso del incremento anterior a su propio peso. Este valor se calcula dividiendo el peso total del tramo por el número de incrementos por tramo. Es decir:

$$Peso(k) = Peso(k - 1) + \frac{Tramo(i, 5)}{NInc(1)}$$

Tramo(i, 5) Peso del tramo.

- *Nota:For Next*. Función del programa Visual Basic que utilizamos a lo largo del programa y que sirve para realizar bucles.

3.4 EXPLICACIÓN RUTINA.

Tomamos el valor de $k = 1$ para que el programa sepa que debe empezar siempre por el primer incremento. Los cálculos debemos de realizarlos siempre desde el primer tramo hasta el último, con lo cual nos movemos desde el primer tramo de torre hasta el último para poder saber cuál es la altura de cada tramo de torre, eso es lo que nos indica el primer *For*. El siguiente *For* nos indica cuantas veces debemos de dividir el tramo de torre en cuestión, para saber la altura real de cada incremento. Por último sumamos a la variable k la unidad, para así poder pasar al siguiente incremento, y realizar nuevamente los cálculos del siguiente incremento.

'Con estas dos variables, sabemos cuántos incrementos tenemos en cada Tramo

$NInc(1) = \text{Int}(100 / (\text{Dat}(1)))$ 'Nº de Incrementos por cada tramo de torre

$NInc(2) = \text{Int}(\text{Dat}(1) + 2) * Cf(1)$ 'Nº de Incrementos Totales

$k = 1$

'División de la Torre

For i = 1 To $\text{Dat}(1)$

For j = 1 To $NInc(1)$

'Cálculo de la altura de cada incremento

$\text{AltInc}(k) = \text{Tramo}(i, 3) / NInc(1)$

'Cálculo del peso de cada incremento

$\text{Peso}(k) = \text{Peso}(k - 1) + \text{Tramo}(i, 5) / NInc(1)$

$k = k + 1$

Next

Next

4 CÁLCULO DE LOS COEFICIENTES DINÁMICOS Y DE EMPUJE DEL VIENTO

Lo primero que debemos de calcular es la fuerza ejercida por el viento sobre la torre. De manera que siguiendo el EUROCODIGO1-Bases de proyecto y acciones en Estructuras – Parte 2-4 : Acciones del viento. En el capítulo 6 “Fuerzas de Viento” tenemos que:

Para estructuras de celosía y para estructuras verticales en ménsula con una relación de esbeltez altura/anchura > 2 y de sección prácticamente constante (por ejemplo edificios altos, chimeneas, torres) la fuerza, F_{wj} , en el incremento de superficie A_j , a la altura z_j es:

$$F_{wj} = q_{ref} \cdot c_e(z_j) \cdot c_d \cdot c_{fj} \cdot A_j$$

Donde:

q_{ref} es la presión media de la velocidad media del incremento

$c_e(z_j)$ es el coeficiente de exposición que tiene en cuenta el terreno y la altura sobre el nivel del suelo

z_j es la altura del centro de gravedad del incremento de superficie A_j

c_d es el coeficiente dinámico que tiene en cuenta la amplificación dinámica

c_{fj} es el coeficiente de fuerza para el incremento de superficie A_j

A_j es la superficie del incremento

En esta rutina calculamos el coeficiente dinámico y el coeficiente de fuerza de nuestra torre. El coeficiente de exposición lo calculamos en la siguiente rutina.

4.1 COEFICIENTE DE FUERZA

Para este apartado se han seguido las indicaciones del EUROCÓDICO 3 - Diseño de estructuras de acero – Parte 3-1: Torres, mástiles y chimeneas. Este coeficiente de empuje del viento, c_f debe de ser calculado según el área resistente en cada tramo de torre.

Según el Anexo B de EN 1993-3-1:

$$c_f = c_{f,S} + c_{f,A}$$

$c_{f,S}$ Es el coeficiente de empuje del viento sobre la sección de estructura desnuda, determinado de acuerdo con el apartado B.2.2 del Anexo B, usando el coeficiente de solidez ϕ , apropiado a la estructura vacía.

$c_{f,A}$ Es el coeficiente de empuje del viento sobre los elementos auxiliares, determinado de acuerdo con el apartado B.2.3 y B.2.4 del Anexo B, según lo apropiado

El coeficiente $c_{f,S}$ sólo es válido para la estructura resistente, es decir, si se tienen elementos auxiliares cogidos a la estructura resistente, se debe calcular un coeficiente de empuje especial $c_{f,A}$, siguiendo las Tablas B.2.1, B.2.2 y el apartado B.2.4 del Anexo B. Sin embargo, cuando las áreas proyectadas de estos elementos auxiliares sobre cada cara no exceden de un 10% del total, se pueden tratar como propios de la estructura, sin la necesidad de calcular más coeficientes de empuje. Sólo es necesario $c_{f,S}$.

En el caso que se trata en este proyecto, las torres albergarán dispositivos de medición de viento, escaleras... siendo éstos, pequeños y diseñados para ofrecer poca resistencia al viento, por lo tanto, nos acogemos a este punto del apartado B.2.1.3 del Anexo B.

Por lo tanto $c_{f,A}$ es 0, y tan solo se debe calcular $c_{f,S}$.

El Eurocódigo expone diferentes maneras para calcular este coeficiente en el apartado B.2.2.1 del anexo B.

En el caso de torres de secciones con planta cuadrada o triangular, el coeficiente total de empuje del viento $c_{f,S}$ de una sección en la dirección del viento es:

$$c_{f,S} = K_{\theta} \cdot c_{f,S,0}$$

$c_{f,S,0}$ Es el coeficiente normal global de empuje (presión) del viento sobre la sección j, sin efectos de extremos, determinado según el apartado B.2.2.2. Los efectos que se contemplan con el coeficiente K_{θ} (coeficiente de incidencia del viento) son por ejemplo, del desprendimiento de la capa límite, por lo tanto, dependen de la propia geometría de los perfiles que conforman la estructura.

El coeficiente de incidencia del viento K_{θ} se puede obtener de:

Para estructuras cuadradas

$$K_{\theta} = 1,0 + K_1 \cdot K_2 \cdot \sin^2 2\theta$$

Con:

$$K_1 = \frac{0,55 \cdot A_f}{A_s} + \frac{0,8 \cdot (A_c + A_{c,sup})}{A_s}$$

$$\begin{aligned}
 K_2 &= 0,2 \text{ para } 0 \leq \varphi \leq 0,2 \text{ y } 0,8 \leq \varphi \leq 1,0 \\
 &= \varphi \text{ para } 0,2 \leq \varphi \leq 0,5 \\
 &= 1 - \varphi \text{ para } 0,5 \leq \varphi \leq 0,8
 \end{aligned}$$

En los cuales:

θ Es el ángulo de incidencia del viento respecto a la normal a cara en planta. 45°

φ Es el coeficiente o relación de solidez.

A_f Es el área total proyectada, vista perpendicularmente, a la cara de los elementos con sección de lados planos en la cara.

A_c Es el área total proyectada vista perpendicularmente a la cara, de los elementos de sección circular, que se encuentran en regímenes subcríticos. Como no tenemos elementos circulares será nula esta superficie.

$A_{c,sup}$ Es el área total proyectada vista perpendicularmente a la cara, de los elementos con sección circular, que se encuentran en regímenes supercríticos. Como no tenemos elementos circulares será nula esta superficie.

Para el cálculo de $c_{f,S,O}$ se debe usar el apartado B.2.2.2 del Anexo B.

Los valores de los coeficientes normales globales de empuje $c_{f,S,O}$, que son aplicables a una estructura en celosía de sección cuadrada o triangular de lados iguales j , compuesta tanto de elementos de sección de lados planos como de sección circular, se deberían tomar como:

$$c_{f,S,O,j} = c_{f,O,f} \cdot \frac{A_f}{A_s} + c_{f,O,c} \cdot \frac{A_c}{A_s} + c_{f,O,c,sup} \cdot \frac{A_{c,sup}}{A_s}$$

Donde $c_{f,O,f}$, $c_{f,O,c}$ y $c_{f,O,c,sup}$, y son los coeficientes de empuje para secciones compuestas de elementos de sección de lados planos, circular subcrítica y circular supercrítica, respectivamente, dados por:

$$\begin{aligned} c_{f,O,f} &= 1,76 \cdot C_1 \cdot (1 - C_2 \cdot \varphi + \varphi^2) \\ c_{f,O,c} &= C_1 \cdot (1 - C_2 \cdot \varphi) + (C_1 + 0,875) \cdot \varphi^2 \\ c_{f,O,c,sup} &= 1,9 - \sqrt{\{(1 - \varphi) \cdot (2,8 - 1,14 \cdot C_1 + \varphi)\}} \end{aligned}$$

Con C_1 igual a: 2,25 para estructuras cuadradas

1,9 para estructuras triangulares

Con C_2 igual a: 1,5 para estructuras cuadradas

1,4 para las estructuras triangulares

Tenemos que:

$$A_s = A_f + A_c + A_{c,sup}$$

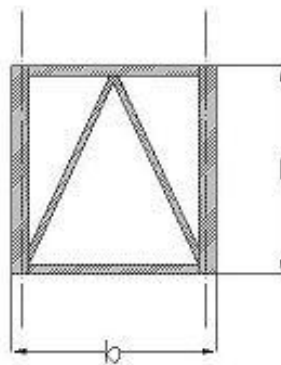
El coeficiente de solidez φ viene dado en EUROCODIGO1-Bases de proyecto y acciones en Estructuras – Parte 2-4 : Acciones del viento por:

$$\varphi = \frac{A}{A_c}$$

Donde:

A es la suma de las superficies proyectadas de los elementos

A_c es la superficie encerrada $A_c = b \cdot l$



A = Área sombreada

Debemos de tener en cuenta que el coeficiente de empuje o fuerza depende de la forma de cada tramo de torre, de manera que tendremos un coeficiente de empuje para cada tramo de torre.

Una vez que se consigue el coeficiente de empuje medio de la torre pasamos al cálculo del coeficiente dinámico.

4.2 COEFICIENTE DINÁMICO

El coeficiente dinámico, c_d , tiene en cuenta los efectos de reducción debidos a la falta de correlación de las presiones sobre las superficies, así como los efectos de mayoración debidos a las componentes de la frecuencia de turbulencia cercanas a la frecuencia fundamental de la estructura.

Se ofrecen dos métodos para calcular las cargas de viento, y con ello el coeficiente dinámico

- El método simplificado se aplica en aquellas estructuras cuyas propiedades estructurales no las hacen susceptibles a la excitación dinámica. Este método puede emplearse también para estructuras levemente dinámicas, utilizando el coeficiente dinámico, c_d . El valor de este coeficiente depende del tipo de estructura, de la altura de la estructura y anchura.

- El método detallado se aplica en aquellas estructuras susceptibles a la excitación dinámica, y para las que el valor del coeficiente dinámico, c_d , es mayor de 1,2.

En nuestro programa utilizaremos el método detallado ya que proporciona siempre resultados más precisos y menos conservadores que el método simplificado. Así que obtendremos valores más precisos del coeficiente dinámico empleando la ecuación (B.2) del anexo B del EUROCODIGO1-Bases de proyecto y acciones en Estructuras – Parte 2-4: Acciones del viento.

El factor dinámico c_d se define por:

$$c_d = \frac{1 + 2 \cdot g \cdot I_v(z_{equ}) \cdot \sqrt{Q_0^2 + R_X^2}}{1 + 7 \cdot I_v(z_{equ})}$$

Donde:

z_{equ} es la altura equivalente de la estructura

$$z_{equ} = 0,6 \cdot h \cdot z_{min}$$

$I_v(z_{equ})$ es la intensidad de turbulencia $I_v(z)$ para $z = z_{equ}$

g es el factor de pico

Q_0 es la respuesta de fondo

R_X es la respuesta resonante

A continuación explicamos cómo obtenemos todos los valores de la ecuación del coeficiente dinámico.

Categorías del terreno

Primeramente debemos de elegir los parámetros de la categoría del terreno. Estos datos se indican en la tabla 8.1

Tabla 8.1. Categorías del terreno y parámetros relativos utilizados en esta Parte.

Categoría del terreno	k_T	$Z_0[m]$	$Z_{min}[m]$	ϵ
I Mar abierto, lagos de al menos 5 km de fetch en la dirección del viento, terreno llano sin obstáculos	0,17	0,01	2	[0,13]
II Granjas con setos, pequeñas estructuras agrarias ocasionales, casas o árboles	0,19	0,05	4	[0,26]
III Áreas suburbanas o industriales, bosques permanentes	0,22	0,3	8	[0,37]
IV Áreas urbanas con al menos el 15% de superficie cubierta con edificios de altura media mayor de 15m	0,24	1	16	[0,46]

Coeficientes para el cálculo de $c_r(z)$

Dada las situaciones normales de este tipo de torres, se selecciona la categoría del terreno I, para realizar los cálculos.

Coeficiente de rugosidad

El coeficiente de rugosidad a una altura z es un valor que se utiliza a menudo durante los cálculos, y se define mediante la siguiente ley logarítmica:

$$c_r(z) = k_T \cdot \ln\left(\frac{z}{z_0}\right) \text{ para } z_{min} \leq z \leq 200 \text{ m}$$

$$c_r(z) = c_r(z_{min}) \text{ para } z \leq z_{min}$$

Donde:

k_T es el factor del terreno

z_0 es la longitud de rugosidad

z_{min} es la altura mínima

Estos valores los obtenemos de la anterior tabla

Coeficiente topográfico

Se toma como coeficiente topográfico por la dificultad de calcularlo según en qué zona se instale el aerogenerador como la unidad

Parámetros del viento y estructurales

La intensidad de turbulencia $I_v(z_{equ})$ se define como:

$$I_v(z_{equ}) = \frac{k_T}{c_t(z_{equ}) \cdot c_r(z_{equ})}$$

El factor de pico g se define como:

$$g = \sqrt{2 \cdot \ln(vt)} + \frac{0,6}{\sqrt{2 \cdot \ln(vt)}}$$

Donde:

$t = 600$ s = tiempo promedio de la velocidad del viento de referencia, v_{ref}

t es la frecuencia esperada, que se define como:

$$v = \sqrt{\frac{v_0^2 \cdot Q_0^2 + n_{1,x}^2 \cdot R_x^2}{Q_0^2 + R_x^2}}$$

$n_{1,x}$ es la frecuencia fundamental, en [Hz], de vibración de la estructura en la dirección del viento (x).

v_0 es la frecuencia esperada, en [Hz], de carga por ráfaga de viento en estructuras rígidas. Se define como:

$$v_0 = \frac{v_m(z_{equ})}{L_i(z_{equ})} \cdot \frac{1}{1,11 \cdot S^{0,615}}$$

Con:

$$S = 0,46 \cdot \left(\frac{b+h}{L_i(z_{equ})} \right) + 10,58 \cdot \left(\frac{\sqrt{b \cdot h}}{L_i(z_{equ})} \right)$$

b, h es la anchura y altura de la estructura

$v_m(z_{equ})$ es la velocidad media del viento $v_m(z)$ para $z = z_{equ}$. Se define como:

$$v_m(z_{equ}) = c_r(z_{equ}) \cdot c_t(z_{equ}) \cdot v_{ref}$$

Donde:

$c_r(z_{equ})$ es el coeficiente de rugosidad

$c_t(z_{equ})$ es el coeficiente topográfico

v_{ref} velocidad del viento de referencia. (Dato que debemos de introducir nosotros)

$L_i(z_{equ})$ es la escala de longitud integral de turbulencia para $z = z_{equ}$. Tenemos tres opciones

$$\begin{array}{lll} L_i(z) = 300 \cdot (z/300)^\varepsilon & (L_i, z \text{ en m}) & \text{para } z_{min} \leq z \leq 300 \text{ m} \\ L_i(z) = 300 \cdot (z_{min}/300)^\varepsilon & (L_i, z \text{ en m}) & \text{para } z \leq z_{min} \\ L_i(z) = 300 \text{ m} & & \text{para } z > 300 \text{ m} \end{array}$$

Nosotros elegiremos la primera opción a la hora de realizar los cálculos.

La respuesta de fondo Q_0 se define como:

$$Q_0^2 = \frac{1}{1 + 0,9 \cdot \left(\frac{b+h}{L_i(z_{\text{equ}})} \right)^{0,63}}$$

La parte de respuesta resonante R_x se define como:

$$R_x^2 = \frac{\pi^2}{2 \cdot \delta} \cdot R_N \cdot R_h \cdot R_b$$

Donde:

δ es el decremento logarítmico de la atenuación de la vibración en la dirección del viento.
 R_N es la función adimensional de la densidad espectral de potencia obtenida de la ecuación:

$$R_N = \frac{n_{1,x} \cdot S_v \cdot (n_x)}{\sigma_v^2} = \frac{6,8 \cdot N_x}{(1 + 10,2 \cdot N_x)^{5/3}}$$

Con:

$$N_x = \frac{n_{1,x} \cdot L_i(z_{\text{equ}})}{v_m(z_{\text{equ}})}$$

La frecuencia fundamental de flexión $n_{1,x}$ se puede estimar así:

$$n_{1,x} = \frac{46}{h}$$

h es la altura de la estructura en [m]

R_h, R_b son las funciones de admitancia aerodinámica para desplazamiento uniforme, y se expresan en términos de la función

$$R_t = \frac{1}{\eta} - \frac{1}{2\eta^2} \cdot (1 - 2^{-2\eta}) \quad \text{para } \eta > 0$$

$$R_t = 1 \quad \text{para } \eta = 0$$

Con:

$$R_h = R_t \quad \text{fijando } \eta = \frac{4,6 \cdot N_{1,x} \cdot h}{L_i(z_{\text{equ}})}$$

$$R_b = R_t \quad \text{fijando } \eta = \frac{4,6 \cdot N_{1,x} \cdot b}{L_i(z_{\text{equ}})}$$

La frecuencia fundamental $n_{1,x}$ y el decremento logarítmico δ los obtenemos de EUROCODIGO1-Bases de proyecto y acciones en Estructuras – Parte 2-4 : Acciones del viento. Anexo C. C.4 Características Dinámicas.

La frecuencia fundamental de flexión n_1 en edificios de varias plantas se puede estimar así:

$$n_1 = \frac{46}{h}$$

Donde:

h es la altura de la estructura en [m]

La misma expresión puede servir de orientación para estructuras de una planta y torres

El decremento logarítmico fundamental del amortiguamiento δ se puede obtener de:

$$\delta = \delta_s + \delta_a + \delta_d$$

Donde:

δ_s es el decremento logarítmico estructural fundamental del amortiguamiento;
 δ_a es el decremento logarítmico aerodinámico fundamental del amortiguamiento;
 δ_d es el decremento logarítmico fundamental del amortiguamiento debido a dispositivos especiales (amortiguadores sintonizados de masa, etc.).

El decremento logarítmico estructural fundamental del amortiguamiento se puede obtener por:

$$\delta_s = a_1 \cdot n_1 + b_1 \geq \delta_{\min}$$

Donde:

n_1 es la frecuencia fundamental de flexión [Hz];
 a_1, b_1, δ_{\min} son los parámetros definidos en la tabla C.8. para distintas tipologías estructurales, δ_{\min} se relaciona con pequeños desplazamientos.

Para torres de celosía de acero soldado tenemos que:

$$\delta_s = 0,05$$

El decremento logarítmico aerodinámico fundamental del amortiguamiento, δ_a , para vibraciones en la dirección del viento se obtiene como:

$$\delta_a = \frac{\rho \cdot b \cdot c_f}{2 \cdot n_{1,x} \cdot m_{1,x}} \cdot v_m(z_{\text{equ}})$$

Donde:

ρ es la masa específica del aire $1,25 \text{ Kg/m}^3$

b es la anchura de la superficie estructural expuesta al viento

c_f es el coeficiente de fuerza medio en la dirección del viento

$v_m(z_{\text{equ}})$ es la velocidad media del viento $v_m(z)$ para la altura equivalente de la estructura

$m_{1,x}$ es la masa equivalente fundamental en la dirección del viento

$n_{1,x}$ es la frecuencia fundamental en la dirección del viento

En estructuras verticales con pequeñas variaciones de la distribución de masa, el valor de m_1 se puede aproximar por el valor medio de m del tercio superior de la estructura h_3 .

Con todos estos datos podemos obtener el valor del coeficiente dinámico de la estructura.

Una vez que hemos explicado cómo obtener los coeficientes dinámicos y de fuerza del viento, aplicaremos las ecuaciones paso por paso para realizar los cálculos.

4.3 EXPLICACIÓN RUTINA.

Primero declaramos las variables. Lo que se intenta es que tengan la misma simbología que en el EUROCODIGO para que no haya problemas de comprensión, y puedan seguirse los pasos de cálculo sin problemas.

Dim h As Single	'Altura total de la torre
Dim Zequ As Double	'Altura equivalente de la estructura
Dim IvZequ As Double	'Intensidad de turbulencia a la altura equivalente Zequ
Dim Qo As Double	'Respuesta de fondo
Dim Rx As Double	'Respuesta resonante
Dim v As Double	'Frecuencia esperada
Dim vo As Double	'Frecuencia esperada en HZ, de carga por ráfaga de viento en estructuras rígidas
Dim VmZequ As Double	'Velocidad media del viento a la altura equivalente Zequ
Dim LiZequ As Double	'Escala de longitud integral de turbulencia para la altura equivalente Zequ
Dim n1x As Double	'Frecuencia fundamental HZ, de carga por ráfaga de viento en estructuras rígidas
Dim Nx As Double	'Variable que se utiliza para calcular la función adimensional de la densidad espectral Rn
Dim Rn As Double	'Función adimensional de la densidad espectral de potencia
Dim Rh As Double	'Función de admitancia aerodinámica
Dim Rb As Double	'Función de admitancia aerodinámica
Dim Delta As Double	'Decremento logarítmico
Dim CrZequ As Double	'Coeficiente de rugosidad para la altura equivalente Zequ
Dim CtZequ As Double	'Coeficiente topográfico para la altura equivalente Zequ
Dim S As Double	'Variable que se utiliza para calcular la frecuencia esperada vo
Dim aux As Double	'Variable \square que se utiliza para calcular las funciones de admitancia Rh, Rb

Dim KDelta As Double	'Coeficiente de incidencia del viento
Dim K1 As Double	'Variable de coeficiente de incidencia
Dim K2 As Double	'Variable de coeficiente de incidencia
Dim CoefSolidez(80) As Double	'Coeficiente de Solidez
Dim C1 As Double	
Dim C2 As Double	
Dim Af As Double	'El área total proyectada, vista perpendicularmente, a la cara de los elementos con sección de lados planos en la cara.
Dim Ase As Double	'Sumatorio de todas las "áreas"
Dim Ac As Double	'El área total proyectada vista perpendicularmente a la cara, de los elementos de sección circular, que se encuentran en regímenes subcríticos
Coeficientes de empuje para secciones compuestas de elementos de sección de lados planos, circular subcrítica y circular supercrítica , respectivamente	
Dim Cfsoj As Double	
Dim Cfof As Double	
Dim Cfoc As Double	
Dim Cfocsup As Double	

Calcularemos primero el coeficiente de fuerza de cada tramo, ya que como hemos explicado antes este coeficiente depende del área resistente de cada tramo, con lo que será distinto en todos ellos.

Para realizar el cálculo nos ayudaremos de un bucle *For*, desde el primer tramo hasta el último y aplicaremos las ecuaciones que hemos explicado antes respecto al coeficiente de empuje. En esta rutina utilizamos otra vez la función *If*, con la cual pondremos las condiciones necesarias para obtener K_2

```

For k = 1 To Dat(1)
  Af=Tramo(k,9)

  Ase = Af + Ac + Acsup

  K1 = 0.55 * Af / Ase + 0.8 * (Ac + Acsup) / Ase

  If 0 < CoefSolidez(k) And CoefSolidez(k) < 0.2 Then K2 = 2

  If 0.8 < CoefSolidez(k) And CoefSolidez(k) < 1 Then K2 = 2

  If 0.2 < CoefSolidez(k) And CoefSolidez(k) < 0.5 Then K2 = CoefSolidez(k)

  If 0.5 < CoefSolidez(k) And CoefSolidez(k) < 0.8 Then K2 = 1 - CoefSolidez(k)

  KDelta = 1 + K1 * K2 * (Sin(45 * Pi / 180) * Cos(45 * Pi / 180)) ^ 2

  C1 = 2.25 'Rectangular

```

Javier Vergara

$C2 = 1.5 \text{ 'Rectangular}$

$Cf_{of} = 1.76 * C1 * (1 - C2 * CoefSolidez(k) + CoefSolidez(k) ^ 2)$

$Cf_{oc} = C1 * (1 - C2 * CoefSolidez(k)) + (C1 + 0.875) + CoefSolidez(k) ^ 2$

$Cf_{soj} = Cf_{of} * Af / Ase$

$CF(k) = K\Delta * Cf_{soj}$

Next

Calculamos el coeficiente medio simplemente empelando un *For*, desde el primer hasta el último tramo. Luego dividimos el valor por el número de tramos.

$Cf_{Medio} = 0$

For i = 1 To Dat(1)

$Cf_{Medio} = Cf_{Medio} + CF(i)$

Next

$Cf_{Medio} = Cf_{Medio} / Dat(1)$

Vamos ahora con el coeficiente dinámico.

Calculamos primeramente la altura de toda la estructura para obtener la altura equivalente z_{equ} . Para ello me sirvo de un bucle, que empieza desde el primer tramo hasta el último y donde en la variable h vamos sumando la altura de cada tramo. Ponemos a cero la variable h para que no haya problemas.

$h = 0$

‘ Calcularemos la altura total de la torre

For i = 1 To Dat(1)

$h = h + Tramo(i, 3)$

Next

$Tramo(i, 3) = \text{Altura del Tramo } i$

Una vez obtenida la altura total de la estructura, aplicando la ecuación requerida tenemos la altura equivalente de la estructura. A partir de ahí realizamos los cálculos antes citados

$$Z_{equ} = 0.6 * h * Z_{min}$$

Parámetros del viento y estructurales

$I_v(Z_{equ})$ La intensidad de turbulencia para la altura equivalente de la estructura

$CrZ_{equ} = K_t * \log(Z_{equ} / Z_o)$ 'Coeficiente de rugosidad

$$CtZ_{equ} = 1$$

$$I_vZ_{equ} = K_t / (CrZ_{equ} * CtZ_{equ})$$

Factor de pico g. Debemos de calcular varios valores para poder llegar hasta el factor de pico g

$$LiZ_{equ} = 300 * (Z_{equ} / 300)^e$$

$$VmZ_{equ} = Dat(20) * CrZ_{equ} * CtZ_{equ}$$

$$S = 0.46 * (Ancho + h) / LiZ_{equ} + 10.58 * ((Ancho * h)^{(1/2)}) / LiZ_{equ}$$

$$Q_o = (1 / (1 + 0.9 * ((Ancho + h) / LiZ_{equ})^{0.63}))^{(1/2)}$$

$$v_o = VmZ_{equ} / LiZ_{equ} * 1 / (1.11 * S^{0.615})$$

Frecuencia Fundamental. ANEXO C.4 - Características Dinámicas

$$n1x = h / 46$$

$$N_x = n1x * LiZ_{equ} / VmZ_{equ}$$

$$R_n = 6.8 * N_x / (1 + 10.2 * N_x)^{(5/3)}$$

Debemos de calcular la masa equivalente fundamental por unidad de longitud.

Llamamos al procedimiento “CálculoMasaEquivalente”. En el calculamos el peso medio del tercio superior de la estructura.

Primero se calcula la altura total de la estructura mediante un bucle *For*, desde el primer hasta el último tramo. A continuación con dos bucles *For*, iremos comparando que incremento se encuentra justo debajo del tercio de la totalidad de la estructura. En ese incremento tendremos el valor de la masa que estamos buscando.

Dim masa As Single

Dim DistanciaInc(100) As Double

h = 0

k = 1

'Calculamos primero la altura de la torre

Javier Vergara

```
For i = 1 To Dat(1)
```

```
    h = h + Tramo(i, 3)
```

```
Next
```

```
h = h / 3
```

```
'Limpiamos el valor DistanciaInc(100)
```

```
For i = 1 To 100
```

```
    DistanciaInc(i) = 0
```

```
Next
```

```
'Calculamos distancia desde cada incremento hasta la punta de la torre,
```

```
'para poder asi calcular, el tercio de la altura de la torre, y saber a que
```

```
'incremento corresponde, para hallar la masa por unidad de longitud
```

```
For i = 1 To Dat(1)
```

```
    For j = 1 To NInc(1)
```

```
        DistanciaInc(k) = DistanciaInc(k - 1) + AltInc(k)
```

```
        If DistanciaInc(k) > h Then
```

```
            If DistanciaInc(k - 1) < h Then
```

```
                masa = Peso(k)
```

```
                GoTo sigue
```

```
            End If
```

```
        End If
```

```
        k = k + 1
```

```
    Next
```

```
Next
```

```
sigue:
```

```
m1x = masa / 2
```

Una vez calculada la masa equivalente, proseguimos con el cálculo. Calculamos el decremento logarítmico.

```
'Calculo del decremento logarítmico
```

Javier Vergara

Deltas = 0.05 "Tabla C.8

$\Delta_{ta} = (1.25 * \text{Ancho} * C_{fMedio}) / (2 * n_{1x} * m_{1x}) * V_{mZequ}$

$\Delta = \Delta_{tas} + \Delta_{ta}$

A la hora de calcular las funciones de admitancia aerodinámica R_h y R_t veíamos como dependían del valor de η . Por ello para calcular estos dos valores utilizamos la función *If – End If* de Visual Basic. En ella pondremos como primera condición que $\eta > 0$ dando un valor a R_h . Si no la cumpliera el programa se salta ese paso y a continuación prueba con la siguiente condición $\eta = 0$ con el que obtenemos otro valor de R_h . Repetimos esta acción para R_t .

Calcularemos primero R_h

$\text{aux} = 4.6 * N_x * h / L_{iZequ}$

If $\text{aux} > 0$ Then

$R_h = 1 / \text{aux} - 1 / (2 * \text{aux}^2) * (1 - \text{Exp}(-2 * \text{aux}))$

End If

If $\text{aux} = 0$ Then

$R_h = 1$

End If

a continuación R_b

$\text{aux} = 4.6 * N_x * \text{Ancho} / L_{iZequ}$

If $\text{aux} > 0$ Then

$R_b = 1 / \text{aux} - 1 / (2 * \text{aux}^2) * (1 - \text{Exp}(-2 * \text{aux}))$

End If

If $\text{Int}(\text{aux}) = 0$ Then

$R_b = 1$

End If

$R_x = (\pi^2 / (2 * \Delta)) * R_n * R_h * R_b^{(1/2)}$

$v = ((v_o^2 * Q_o^2 + n_{1x}^2 * R_x^2) / (Q_o^2 + R_x^2))^{(1/2)}$

$g = (2 * \text{Log}(v * 600))^{(1/2)} + 0.6 / (2 * \text{Log}(v * 600))^{(1/2)}$

'Coeficiente dinámico

$C_d = (1 + 2 * g * I_{vZequ} * (Q_o^2 + R_x^2)^{(1/2)}) / (1 + 7 * I_{vZequ})$

5 CÁLCULO DE LA FUERZA DE VIENTO EJERCIDA EN CADA INCREMENTO

Una vez calculados los coeficientes dinámico y de fuerza, la fuerza, F_{wj} , en el incremento de superficie A_j , a la altura z_j es:

$$F_{wj} = q_m(z) \cdot c_e(z_j) \cdot c_d \cdot c_{fj} \cdot A_j$$

De manera que debemos de calcular tanto el coeficiente de exposición $c_e(z_j)$ como la presión media relativa al incremento $q_m(z)$.

Para encontrar las cargas de viento. EUROCÓDICO 3 - Diseño de estructuras de acero – Parte 3-1: Torres, mástiles y chimeneas remite a la norma nacional para encontrar la velocidad de referencia, sin embargo, estas acciones suelen ser marcadas por el cliente. Pero nuestro programa las calcula.

5.1 CÁLCULO DE LA FUERZA

La velocidad básica del viento v_{ref} , (que es la que todas las normas de viento de Europa suelen dar), es la velocidad media del viento medida durante 10 min, independientemente de la dirección del viento y la época del año, a 10 m sobre el nivel del suelo, en campo abierto, con vegetación baja como la hierba, y con obstáculos aislados con una separación de al menos 20 veces la altura de los obstáculos. Como se ha especificado en el apartado de introducción de datos, nosotros introduciremos este valor.

Mediante este dato y otros que se van especificando, se calcula la velocidad media del viento $v_m(z)$ a una altura z por encima del terreno. En nuestros cálculos lo que hacemos es encontrar una velocidad media para cada incremento, ya que la distribución del viento es realmente parabólica.

La velocidad media se define como:

$$v_m(z) = c_r(z) \cdot c_t(z) \cdot v_{ref}$$

Donde:

$c_r(z)$ es el coeficiente de rugosidad explicado en el apartado anterior.

$c_t(z)$ es el coeficiente topográfico explicado en el apartado anterior.

v_{ref} es la velocidad del viento de referencia, dato que se introduce manualmente.

El coeficiente de exposición, $c_e(z)$, tiene en cuenta los efectos que la rugosidad del terreno, la topografía y la altura sobre el nivel del suelo, provocan sobre la velocidad media del viento y turbulencia. Se define por:

$$c_e(z) = c_r^2(z) \cdot c_t^2(z) \cdot [1 + 2 \cdot g \cdot l_v(z)]$$

Donde:

g es el factor de pico, calculado en el apartado anterior.

$l_v(z)$ es la intensidad de turbulencia, dada por:

$$l_v(z) = \frac{k_T}{c_r(z) \cdot c_t(z)}$$

Para cada incremento calculamos un coeficiente de rugosidad, topográfico y de exposición.

Una vez que tenemos la velocidad media del incremento $v_m(z)$ la presión media de cada incremento $q_m(z)$ se obtendrá de:

$$q_m(z) = \frac{\rho}{2} \cdot v_m(z)^2$$

Donde:

$v_m(z)$ es la velocidad media del viento en cada incremento
 ρ es la densidad del aire, tendrá un valor de 1,25 Kg/m³

Con todo esto podemos calcular por fin la fuerza que ejerce el viento en cada incremento. La fuerza ejercida estará en Newtons.

Para calcular la fuerza sobre las aspas del rotor aplicaremos la ecuación.

$$F = q_m(z) \cdot A$$

A Es el área resistente igual a

$$A = \pi \cdot \left(\frac{d}{2}\right)^2$$

d Diámetro del rotor de las aspas

5.2 EXPLICACIÓN RUTINA.

Declaración de variables.

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim h As Double 'Variable que se utiliza para calcular la altura respecto del suelo de cada incremento

Calculamos primeramente la altura de toda la estructura para obtener la altura total de la estructura. Para ello me sirvo de un bucle, que empieza desde el primer tramo hasta el último y donde en la variable h vamos sumando la altura de cada tramo. Ponemos a cero la variable h para que no haya problemas. Con este valor de h, calcularemos la altura de cada incremento respecto del suelo.

h = 0

'Calcularemos la altura total de la torre

For i = 1 To Dat(1)

h = h + Tramo(i, 3)

Next

En esta segunda orden calculamos tanto el coeficiente de rugosidad, topográfico y de exposición de cada incremento. Así como la velocidad media y presión media. Por último la fuerza que ejerce el viento en la torre para cada incremento. Esta fuerza la guardaremos en la variable FVientoCaso1(k).

Calculamos también la fuerza que ejerce el viento sobre la superficie de la góndola y sobre el rotor. Estos dos valores los guardaremos en una nueva variable FVientoElementos(2).

Para calcular la fuerza sobre las aspas aplicamos las ecuaciones.

$$q_m(z) = \frac{\rho}{2} \cdot v_m(z)^2$$

$$F_{aspas} = q_m(z) \cdot A_j$$

La densidad media $q_m(z)$ para la fuerza del viento en góndola y aspas será la misma y se calculará.

En cuanto al área resistente de las aspas se calcula con el diámetro que introducimos en la variable $Dat(8)$.

Tomamos el valor de $k = 1$ para que el programa sepa que debe empezar siempre por el primer incremento. Los cálculos debemos de realizarlos siempre desde el primer tramo hasta el último, con lo cual nos moveremos desde el primer tramo de torre hasta el último para poder saber cuál es el área resistente y el coeficiente de fuerza da cada uno de ellos. El coeficiente de fuerza lo hemos calculado en el anterior apartado “4 Cálculo de los coeficientes dinámicos y de empuje del viento”.

Todo ello nos lo indica el primer For. El siguiente For nos indica cuantas veces debemos de realizar los cálculos para cada tramo de torre.

En este segundo For es donde realizamos los cálculos.

Lo primero que hacemos es calcular la altura del incremento respecto del suelo.

Suponemos que tomamos el centro de cada incremento como punto de incidencia de la fuerza. Para calcularlo simplemente restaremos a la variable h la mitad de la altura del incremento. La altura de cada incremento se calculó en la primera rutina “División de incrementos”. A continuación se calculan, como se ha explicado anteriormente, el coeficiente de rugosidad, el coeficiente de topográfico, la intensidad de turbulencia, velocidad media en incremento, presión media en incremento, fuerza del viento en cada incremento.

Todas estas variables son vectores de 100 espacios, para poder guardar en cada espacio el valor de cada incremento. Por último, a la variable h le restaremos el valor de la altura de cada incremento para poder así calcular correctamente la altura respecto del suelo del siguiente incremento. Y sumaremos a la variable k la unidad, para así poder pasar al siguiente incremento, y realizar nuevamente los cálculos del siguiente incremento.

Para calcular la fuerza del viento del Caso 2 simplemente multiplicaremos la fuerza del viento del Caso 1 por 1.2. La diferencia entre estas dos fuerzas es que el viento incide sobre una mayor superficie, con lo que estimamos que multiplicarlo por 1.2 es un coeficiente apropiado.

'Calculamos la fuerza que el viento ejerce sobre la superficie de la góndola

'Coeficiente de rugosidad

$$Cr(0) = Kt * \text{Log}((h + Dat(7) / 2) / Zo)$$

'Coeficiente topográfico

$$Ct(0) = 1$$

'Intensidad de turbulencia

$$I_v(0) = K_t / (C_r(0) * C_t(0))$$

'Coeficiente de exposición

$$C_e(0) = C_r(0) ^ 2 * C_t(0) ^ 2 * (1 + 2 * g * I_v(0))$$

'Velocidad media incremento'

$$V_m(0) = Dat(9) * C_r(0) * C_t(0)$$

'Presión en cada incremento

$$P_r(0) = 1.25 * 0.5 * V_m(0) ^ 2$$

'Fuerza del viento en cada incremento

$$F_{VientoElementos}(1) = P_r(0) * C_e(0) * CF(1) * C_d * Dat(6)$$

$$F_{VientoElementos}(2) = P_r(0) * (Dat(8)/2) ^ 2$$

Calcularemos, el coeficiente de rugosidad, topográfico y de exposición de cada incremento.

Así como la velocidad y presión media. Por último la fuerza que ejerce el viento en la torre para cada incremento

k = 1

For i = 1 To Dat(1)

For j = 1 To NInc(1)

$$AltViento(k) = h - AltInc(k) / 2$$

'Coeficiente de rugosidad

If AltViento(k) > Zmin Then

$$C_r(k) = K_t * Log(AltViento(k) / Z_o)$$

Else

$$C_r(k) = K_t * Log(Zmin / Z_o)$$

End If

'Coeficiente topográfico

$$C_t(k) = 1$$

'Intensidad de turbulencia

$$I_v(k) = K_t / (C_r(k) * C_t(k))$$

'Coeficiente de exposición

$$C_e(k) = C_r(k) ^ 2 * C_t(k) ^ 2 * (1 + 2 * g * I_v(k))$$

'Velocidad media incremento'

$$V_m(k) = Dat(9) * C_r(k) * C_t(k)$$

'Presión en cada incremento

$$P_r(k) = 1.25 * 0.5 * V_m(k) ^ 2$$

'Fuerza del viento en cada incremento

$$F_{VientoCaso1}(k) = P_r(k) * C_e(k) * C_d * CF(i) * Tramo(i, 9) / (NInc(1))$$

$FVientoCaso2(k) = FVientoCaso1(k) * 1.2$

$h = h - AltInc(k)$

$k = k + 1$

Next

Next

6 CÁLCULO DE LOS MOMENTOS FLECTORES, FUERZAS CORTANTES Y PESOS

En este apartado calculamos el valor de los momentos flectores, fuerzas cortantes y pesos de cada uno de los incrementos de la torre.

Al tener distintas solicitaciones sobre la torre, calculamos momentos independientes para cada una de estas solicitaciones. Guardamos estos valores en diferentes espacios de memoria dentro de una variable. Las solicitaciones son 4 de manera que tendremos 4 tipos de momentos que depende de:

- Momentos debidos al viento en la góndola
- Momentos debidos al viento en las aspas
- Momentos debido al peso descentrado de la parte superior que forman la góndola y aspas.
- Momentos debidos al viento en los tramos de torre.

Como hay dos casos de estudio tendremos que crear dos variables para poder guardar estos momentos. Guardamos todos estos momentos en dos nuevas variables MomentosCaso1(100,4) y MomentosCaso2(100,4). Para un incremento k lo momentos quedan almacenados en la matriz de la variable:

- Momentos debidos al viento en la góndola Caso 1, MomentosCaso1(k,1)
- Momentos debidos al viento en las aspas Caso1, MomentosCaso1(k,2)
- Momentos debidos al peso de la parte superior que forman la góndola y aspas Caso 1, MomentosCaso1(k,3)
- Momentos debidos al viento en los tramos de torre Caso 1, MomentosCaso1(k,4)
- Momentos debidos al viento en la góndola Caso 2, MomentosCaso2(k,1)
- Momentos debidos al viento en las aspas Caso2, MomentosCaso2(k,2)
- Momentos debidos al peso de la parte superior que forman la góndola y aspas Caso 2, MomentosCaso2(k,3)
- Momentos debidos al viento en los tramos de torre Caso 2, MomentosCaso2(k,4)

Para almacenar los pesos de los incrementos de tramos por encima de un incremento k, creamos una nueva variable *Peso*(100).

Explicaremos cada uno de los pasos dados. Pondremos como ejemplo el Caso 1. Para el Caso 2 es exactamente igual.

6.1 CÁLCULO DE MOMENTOS DEBIDOS AL VIENTO EN LA GÓNDOLA

Para calcular los momentos de cada incremento, cortamos el incremento por su punto más bajo y calculamos los momentos respecto de ese punto que producen todas las fuerzas por encima de él. Empezamos con la resistencia que produce la góndola frente al viento.

El viento siempre incide sobre la misma superficie de la góndola, sin importar si estudiamos el Caso 1 o Caso 2, ya que toda la parte superior, góndola y aspas, giran en la dirección del viento.

La fuerza que el viento ejerce en la góndola crea un momento en cada uno de los incrementos de la torre.

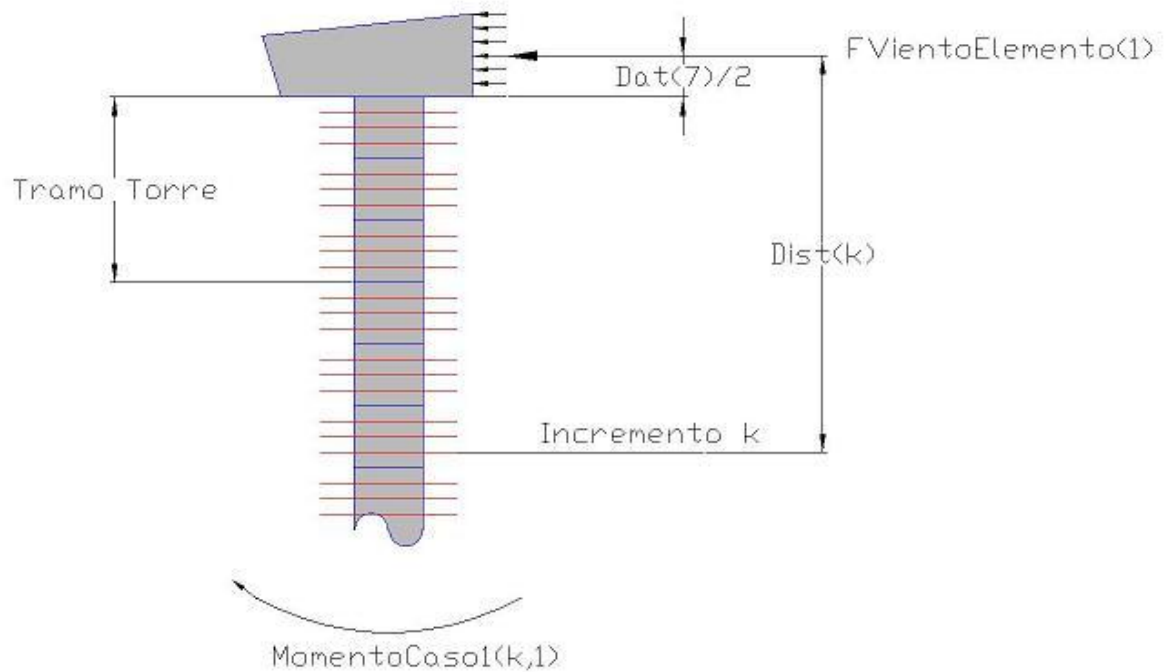
Suponemos que la fuerza del viento se reparte equitativamente por toda la superficie frontal de la góndola, de manera que su punto de aplicación esta justo a la mitad de la altura de la góndola. Lo más correcto sería dividir la góndola también incrementos pero la estimación de que la fuerza del viento esta aplicada en el centro de la góndola es adecuada. En la figura queda constancia de ello.

A esta distancia hemos de sumarle la distancia del punto más bajo del incremento. Por ello utilizamos una nueva variable $Dist(100)$, en el que iremos sumando esta distancia. Para un incremento k es $Dist(k)$. Conclusión el momento causado por el viento en la góndola en un incremento k es:

$$MomentoCaso1(k, 1) = FVientoElementos(1) \cdot Dist(k)$$

La fuerza que el viento ejerce sobre la góndola $FVientoElementos(1)$ se ha calculado en el apartado anterior “5 Cálculo de la fuerza de viento ejercida en cada incremento”.

Asignamos un valor fijo a todo el incremento, cuando en realidad va variando dentro de un intervalo de valores, directamente proporcional a la distancia, pero al elegir la distancia máxima el error cometido será siempre por exceso, por lo cual estamos introduciendo un pequeño coeficiente de seguridad.

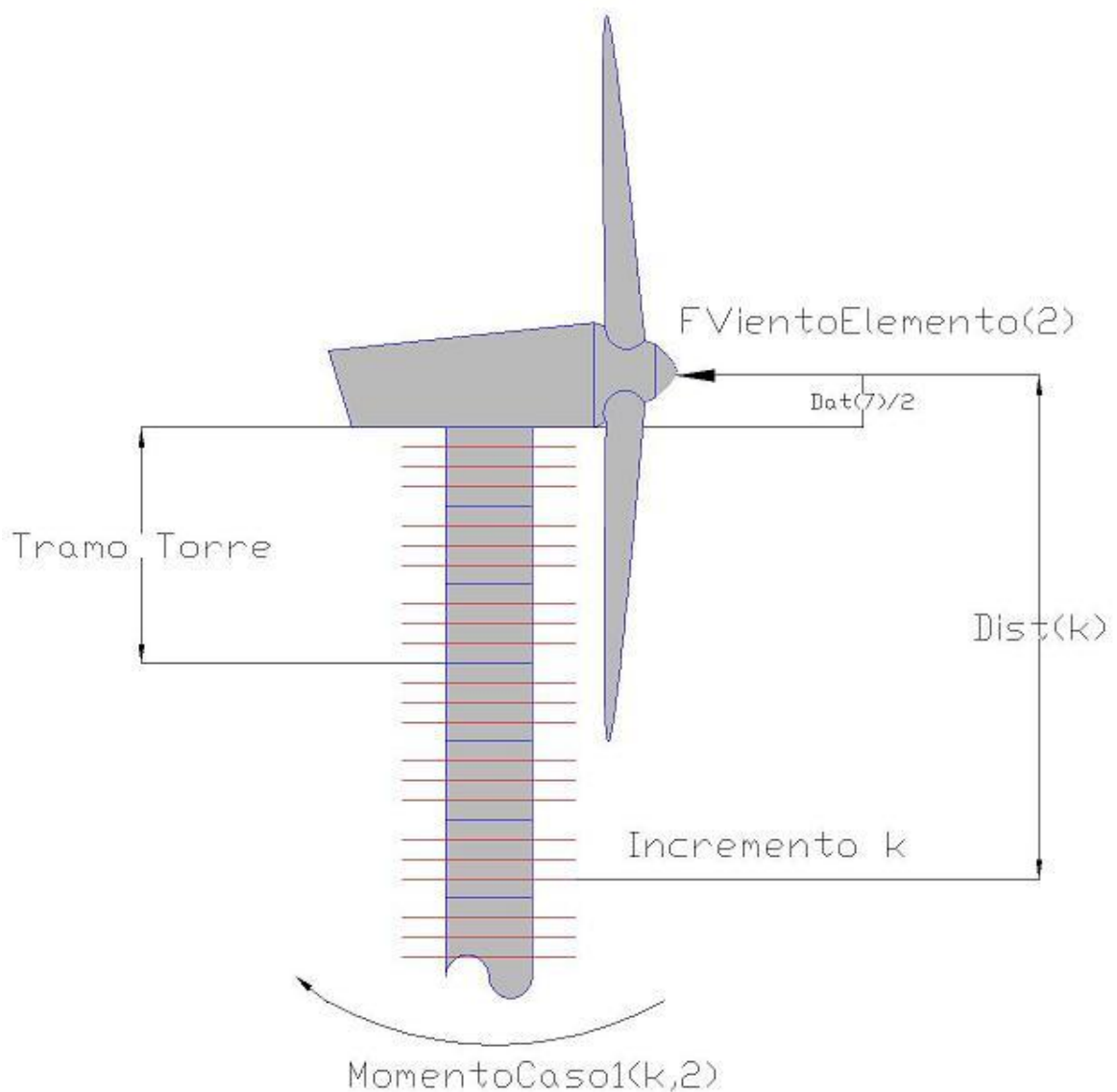


6.2 CÁLCULOS DE MOMENTOS DEBIDOS AL VIENTO EN ASPAS.

El cálculo de los momentos causados por el viento en aspas es idéntico al de los momentos causados por el viento en la góndola.

La fuerza que el viento ejerce sobre las aspas $F_{VientoElementos(2)}$ se ha calculado en el apartado anterior “5 Cálculo de la fuerza de viento ejercida en cada incremento”.

$$MomentoCaso1(k, 2) = F_{VientoElementos(2)} \cdot Dist(k)$$



6.3 CÁLCULO DE MOMENTOS DEBIDOS AL PESO DESCENTRADO DE GÓNDOLA Y ASPAS.

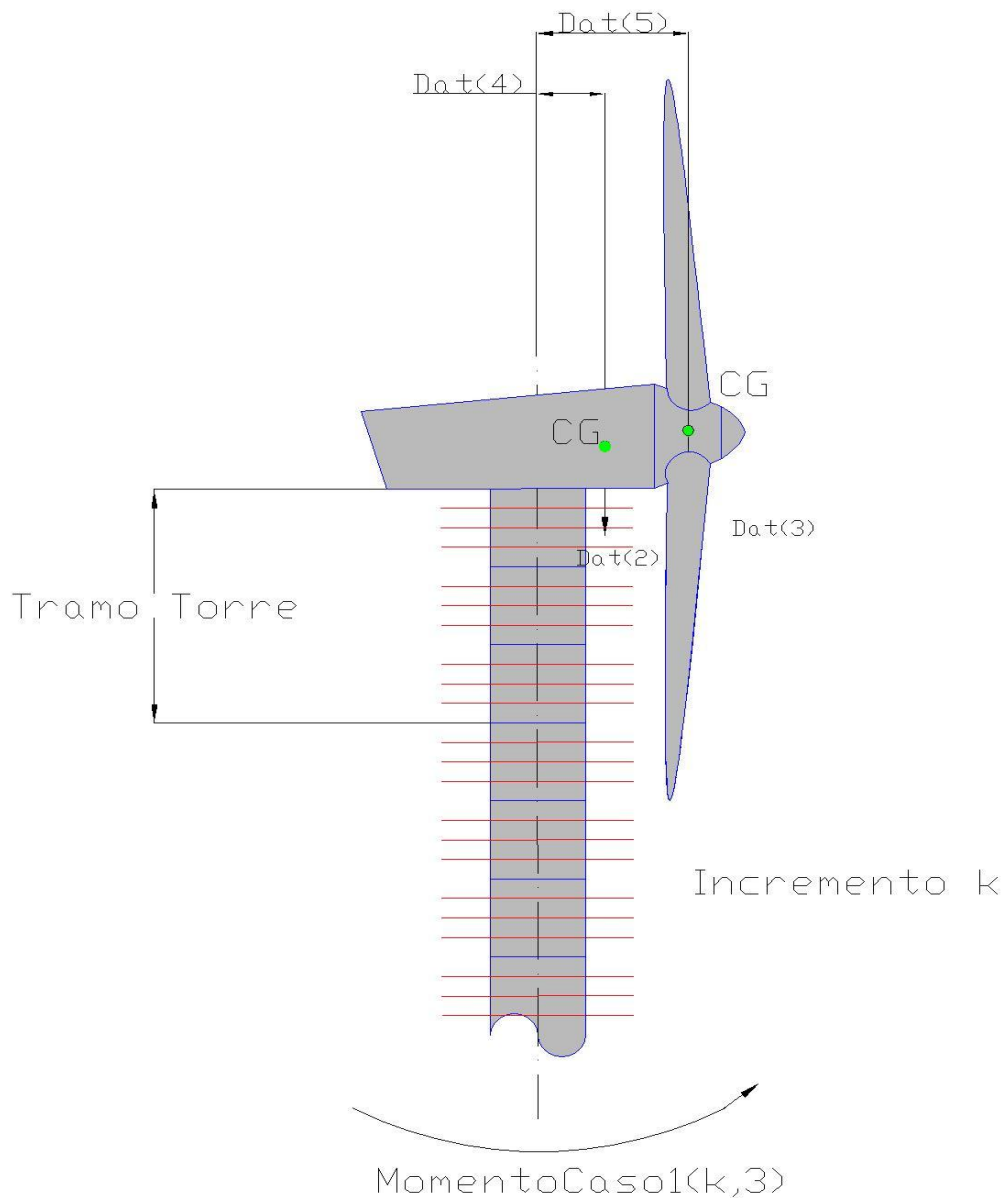
Los elementos de la parte superior, son la góndola y aspas. Estos elementos están descentrados respecto al centro de la torre, de manera que generan unos momentos. Puede que no sean muy desfavorables pero debemos de tenerlos en cuenta. Este momento es constante a lo largo de la torre, de manera que no ofrece ningún problema. Asignamos a todos los incrementos tanto para el Caso 1 como para el Caso 2 este valor. Estos nuevos momentos son de sentido contrario respecto a los calculados anteriormente, de manera que luego al realizar los cálculos de la estructura deberemos de restarlos.

Calculamos primeramente el valor de este momento. Simplemente aplicando los pesos por la distancia al centro de la torre obtenemos el valor del momento. Tenemos una nueva variable *MomentoPeso*. Lo multiplicamos por 10 para tener Newtons como unidad.

$$\text{MomentoPeso} = (\text{Dat}(2) \cdot \text{Dat}(4) + \text{Dat}(3) \cdot \text{Dat}(5)) \cdot 10$$

Como hemos dicho para cada incremento daremos este valor.

$$\text{MomentoCaso1}(k,3) = \text{MomentoPeso}$$



6.4 CÁLCULO DE MOMENTOS DEBIDOS AL VIENTO EN TRAMOS DE TORRE.

Para el cálculo de estos momentos utilizamos la fuerza que el viento ejerce en cada incremento y que hemos calculado en el apartado “5 Cálculo de la fuerza de viento ejercida en cada incremento”. Suponemos que la fuerza del viento incide sobre todo el incremento equitativamente, con lo que la fuerza está ejercida en la mitad del incremento. Todos los cálculos de los momentos se harán respecto a la base del incremento, es decir el punto más alejado posible. Con lo que el error que cometemos es por exceso.

Son necesarios 3 bucles de cálculo. Uno desde el primer tramo de torre hasta el último. Con él nos movemos de un tramo de torre a otro. En el segundo bucle nos movemos desde el primer incremento de cada tramo hasta el último. En estos dos primeros bucles calculamos el momento que ejerce la fuerza del viento en ese mismo incremento. Para conseguirlo multiplicamos la fuerza del viento por la distancia hasta la base del incremento.

$$MomentosCaso1(k, 4) = FVientoCaso1(k) \cdot DistBucle$$

Siendo:

$$DistBucle = \frac{AltInc(k)}{2}$$

AltInc(k) Altura del incremento *k*. Lo hemos calculado en el apartado “3 División de la torre en incrementos”

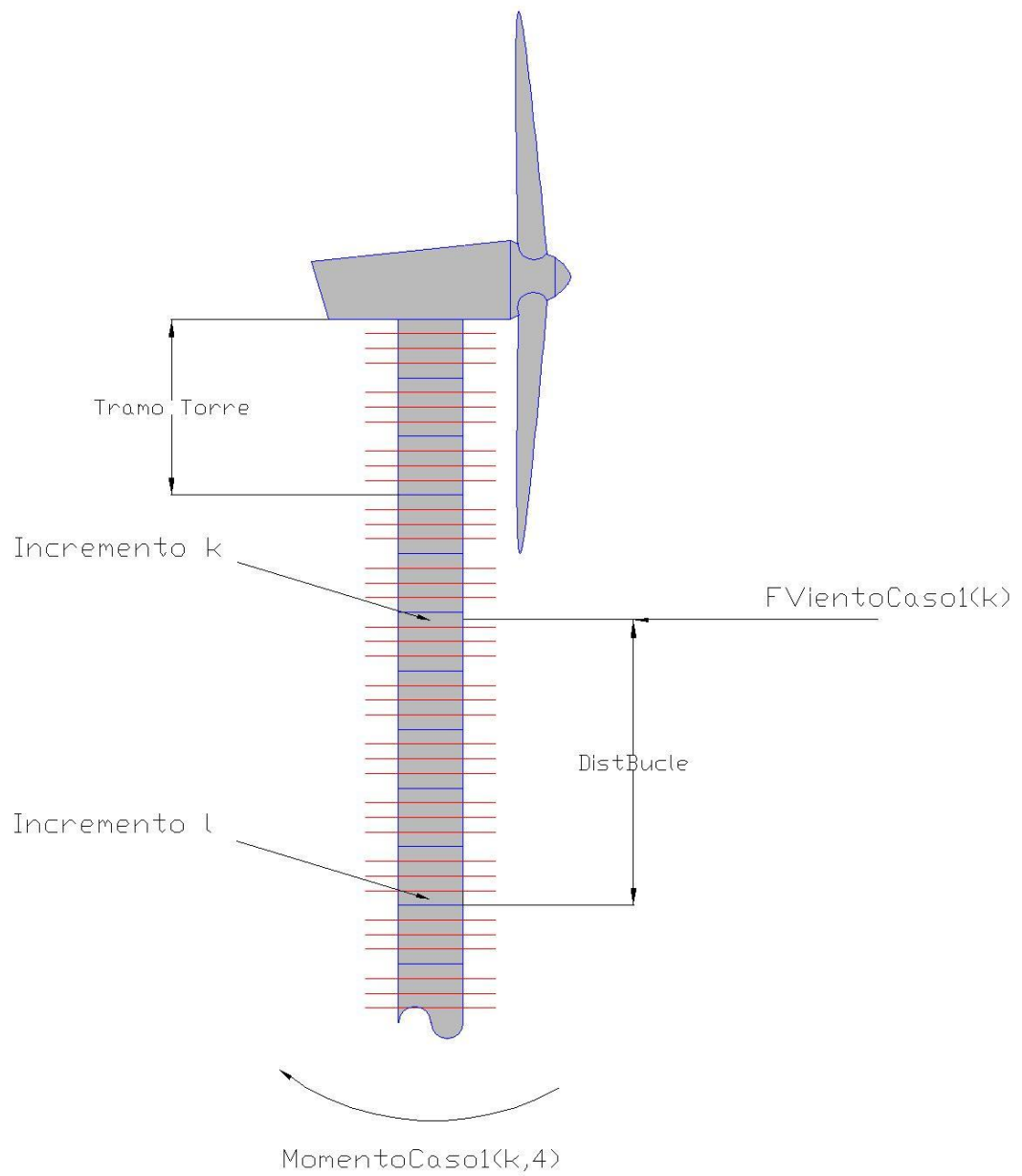
Pero deberemos de sumarle el valor que ya se haya calculado de esos incrementos. Con la explicación del tercer y último bucle todo queda más claro.

$$MomentosCaso1(k, 4) = MomentosCaso1(k, 4) + FVientoCaso1(k) \cdot DistBucle$$

En el tercer bucle calculamos el momento que ejerce la fuerza del incremento que se está estudiando, en todos los incrementos que se encuentran por debajo. Este bucle se mueve desde $k + 1$, es decir el siguiente incremento al que estamos estudiando, hasta el último incremento. La fuerza es constante pero la distancia aumenta en un valor igual a la altura del incremento que se esté estudiando. Esta distancia se guarda en la variable *Distbucle*, que depende del punto de aplicación de la fuerza y el punto del incremento respecto al cual calculamos los momentos.

$$DistBucle = DistBucle + AltInc(l)$$

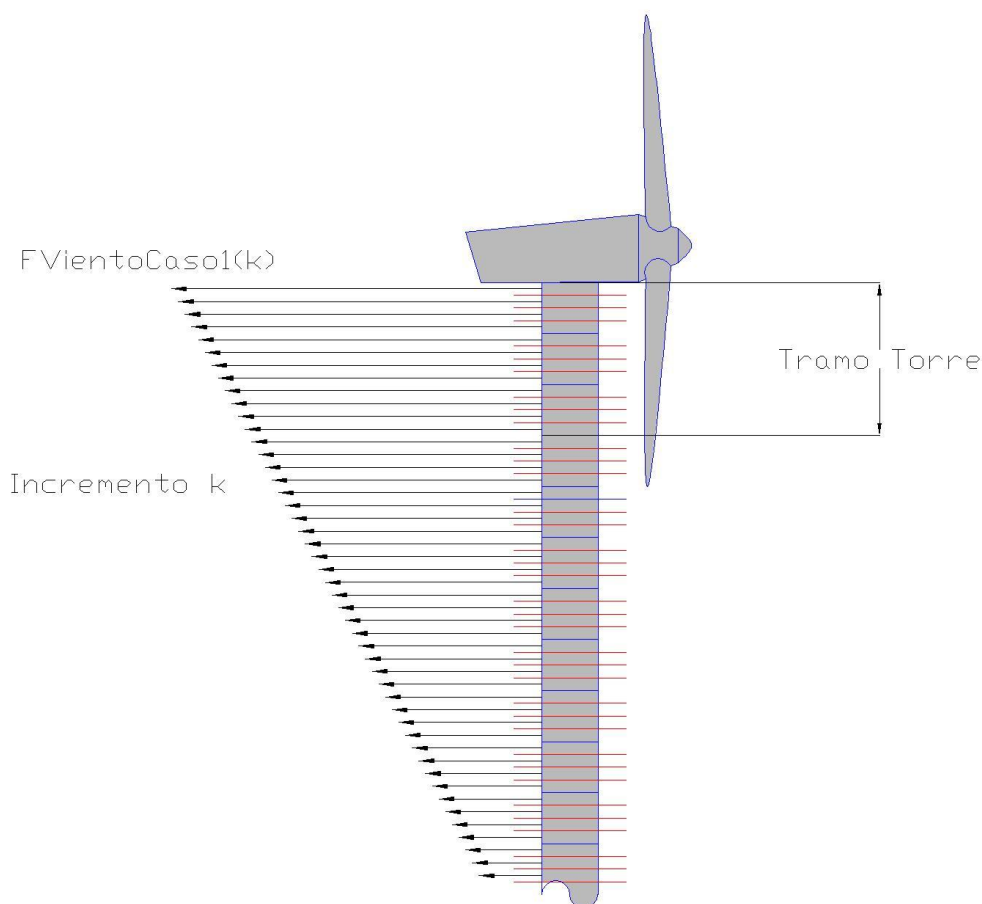
$$MomentosCaso1(l, 4) = MomentosCaso1(k, 4) + FVientoCaso1(k) \cdot DistBucle$$



6.5 FUERZAS CORTANTES DEBIDAS AL VIENTO.

Los valores de fuerza cortante que calculamos, son los debidos al viento en tramos únicamente, por ello los almacenamos en una variable *EsfCortanteVientoCaso1(100)* y *EsfCortanteVientoCaso2(100)*. Para el cálculo de estos esfuerzos utilizamos la fuerza que el viento ejerce en cada incremento y que hemos calculado en el apartado “5 Cálculo de la fuerza de viento ejercida en cada incremento”. Su cálculo es muy sencillo, el valor de la variable *EsfCortanteVientoCaso1(k)* para un incremento k es igual al viento sobre todos los incrementos por encima de él, más el viento propio. Lo conseguimos sumando al viento del incremento propio el valor del viento del incremento anterior. Es decir para un incremento k tenemos que:

$$EsfCortanteVientoCaso1(k) = EsfCortanteVientoCaso1(k - 1) + FVientoCaso1(k)$$



6.6 EXPLICACIÓN RUTINA.

Declaración de variables.

```
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim l As Integer
Dim Dist(100) As Double
Dim DistBucle As Double
```

Lo primero que hacemos es calcular la distancia a la que esta la fuerza del viento en la góndola y las aspas, respecto de la base del elemento superior. Como hemos comentado antes suponemos que la fuerza esta ejercida en el medio de la superficie de la góndola. Con lo que estará a la mitad de la altura.

Calculamos también el momento producido por los pesos descentrados de la góndola y aspas.

'Distancia respecto de la base de la parte superior del punto de incidencia del viento en la góndola y aspas

$\text{Dist}(0) = \text{Dat}(7) / 2$

'Momento Causado por el peso de la góndola y las aspas ya que están "descentrados"

$\text{MomentoPeso} = \text{Dat}(2) * \text{Dat}(4) + \text{Dat}(3) * \text{Dat}(5)$

Utilizamos como hemos comentado antes tres bucles. El primero va desde el primer tramo de torre hasta el último. El segundo desde el primer incremento de cada tramo hasta el último. Dentro de estos dos bucles calculamos los momentos ejercidos por el viento en la góndola, aspas, viento en tramo de torre, y el momento constante debido a los pesos descentrados de la góndola más el del peso de las aspas. En el tercer bucle calculamos el momento que ejerce la fuerza del incremento que se está estudiando, en todos los incrementos que se encuentran por debajo. Al final del bucle calculamos también los esfuerzos cortantes. Todos estos cálculos se realizan para los dos casos de estudio. Utilizamos la variable k para movernos entre los incrementos. De manera que siempre que hayamos realizado un bucle, le sumaremos la unidad, para pasar al estudio del siguiente bucle.

```

k=0

For i = 1 To Dat(1)
  For j = 1 to NInc(1)

    k=k+1
    Dist(k) = Dist(k - 1) + AltInc(k)
    DistBucle = AltInc(k) / 2

    'CASO 1

    'Momento causado por viento en la góndola
    MomentosCaso1(k, 1) = FVientoElementos(1) * Dist(k)
    'Momento causado por viento en aspas
    MomentosCaso1(k, 2) = FVientoElementos(2) * Dist(k)
    'Momento causado por el momento del peso cuasado por la góndola y aspas
    MomentosCaso1(k, 3) = MomentoPeso
    'Momento casuado por viento en torre
    MomentosCaso1(k, 4) = MomentosCaso1(k, 4) + FVientoCaso1(k) * DistBucle

    'CASO 2

    'Momento causado por viento en la góndola
    MomentosCaso2(k, 1) = FVientoElementos(1) * Dist(k)
    'Momento causado por viento en aspas
    MomentosCaso2(k, 1) = FVientoElementos(2) * Dist(k)
    'Momento causado por el momento del peso cuasado por la góndola y aspas
    MomentosCaso2(k, 3) = MomentoPeso
    'Momento casuado por viento en torre
    MomentosCaso2(k, 4) = MomentosCaso2(k, 4) + FVientoCaso2(k) * DistBucle

    'Con este bucle sumaremos a cada "incremento", el momento causado por el viento que esté por encima de cada "incremento"

    For l = k + 1 To NInc(2)
      DistBucle = DistBucle + AltInc(l)
      MomentosCaso1(l, 4) = MomentosCaso1(l, 4) + FVientoCaso1(k) * DistBucle
      MomentosCaso2(l, 4) = MomentosCaso2(l, 4) + FVientoCaso2(k) * DistBucle
    Next

    'Cálculo del esfuerzo cortante de cada incremento
    EsfCortanteVientoCaso1(k) = EsfCortanteVientoCaso1(k - 1) + FVientoCaso1(k)
    EsfCortanteVientoCaso2(k) = EsfCortanteVientoCaso2(k - 1) + FVientoCaso2(k)

  Next

```

Por último calculamos el peso de los incrementos. Utilizamos dos bucles. El primero para movernos desde el primer tramo de torre hasta el último. De esta manera sabemos que peso tiene el tramo. Y el segundo bucle lo utilizamos para movernos desde el

primer incremento de cada tramo hasta el último. Utilizamos la variable k, para poder saber en qué incremento estamos. Sumaremos cada vez hasta variable k la unidad, para poder pasar al siguiente incremento y continuar con los cálculos. Para que no haya problemas ponemos la variable k al principio de todo a cero.

k = 0

For i = 1 To Dat(1)

For j = 1 To NInc(1)

k = k + 1

'Calculo del peso de cada incremento

Peso(k) = Peso(k - 1) + Tramo(i, 5) / NInc(1)

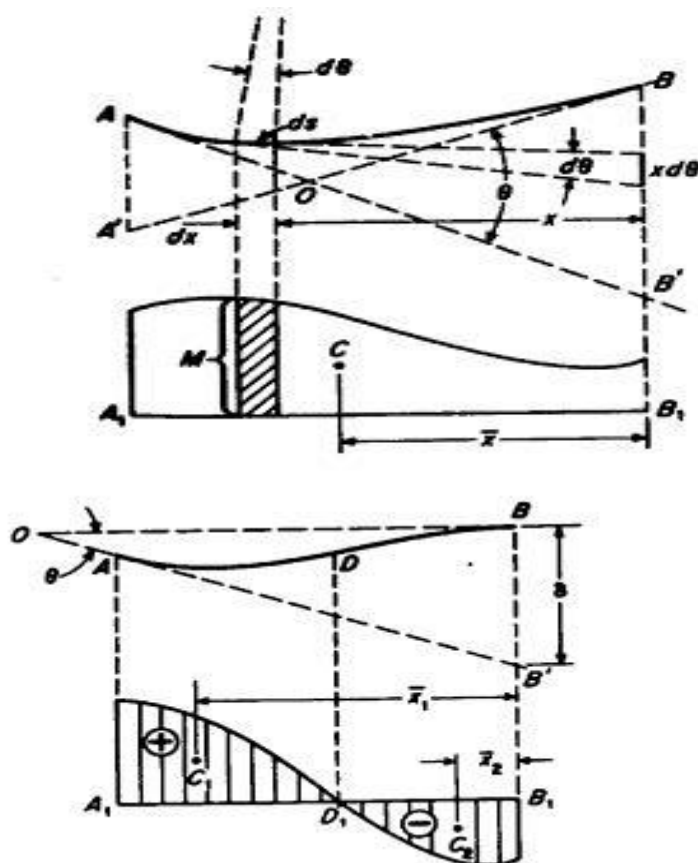
Next

Next

7 CÁLCULO DE LAS FLECHAS DE DEFORMACIÓN CAUSADAS POR LAS DISTINTAS SOLICITACIONES

Para poder calcular las flechas aplicamos el segundo teorema de Mohr. Este teorema dice que dados dos puntos A y B pertenecientes a una línea elástica, la ordenada de B respecto a la tangente en A es igual al momento estático con respecto a B del área de momentos reducidos comprendida entre A y B.

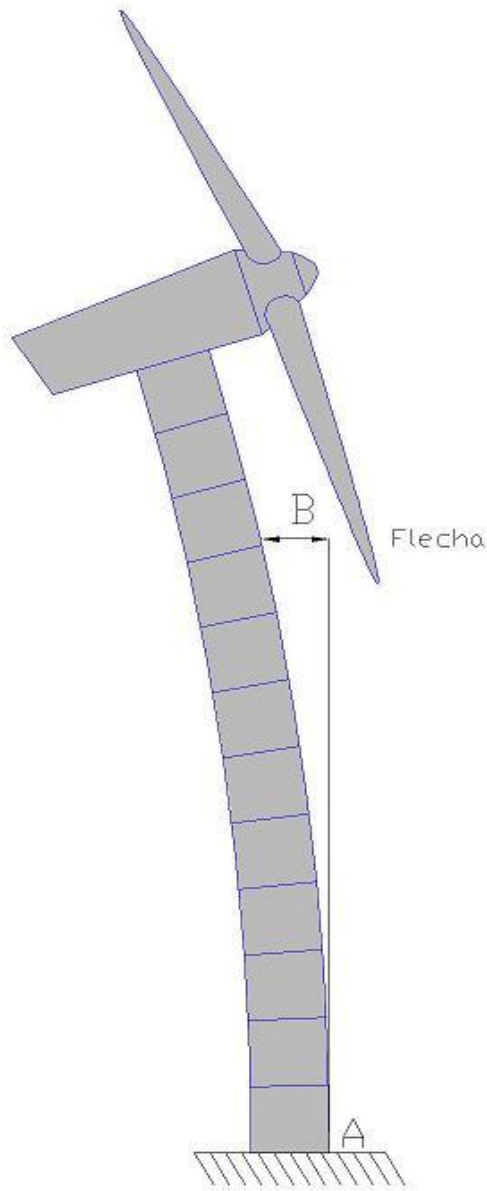
$$\Delta_{B|A} = - \int_{x_A}^{x_B} \frac{M_f(x)}{EI_f} (x - x_A) dx$$



El momento estático recientemente mencionado puede calcularse en forma muy simple multiplicando el área total del diagrama de momentos reducidos comprendida entre A y B por la distancia a su centro de gravedad. Por otro lado, si la figura que representa el diagrama puede descomponerse en figuras elementales tales como rectángulos, triángulos,

parábolas, etc., el momento estático total resultara ser la suma de los correspondientes a cada una de las figuras elementales.

Si tomamos el punto A en la base de la torre, podemos considerarla como una viga empotrada, por lo que la tangente irá en dirección vertical y la flecha a la tangente será la flecha real del punto B.



Los valores de los momentos calculados están N·m, deberemos de dividirlos por $E \cdot I$ para cada incremento, y hacer una reconversión de unidades para obtener las flechas en m.

7.1 CÁLCULO DEL MOMENTO DE INERCIA DE LA SECCIÓN RESPECTO AL EJE TRANSVERSAL Y DIAGONAL.

Lo primero que calculamos es el momento de inercia respecto a los dos casos que estamos suponiendo hasta ahora, es decir respecto al eje y la diagonal.

Al hacer los cálculos suponemos la torre como si fuera una viga, pero en realidad es una estructura metálica en la que las barras que forman la estructura trabajan a tracción o a compresión. Para hacer un estudio correcto habría que realizar un análisis de las deformaciones bastante complejo.

Suponemos que son los montantes del tramo los encargados de contrarrestar los momentos flectores que aparecen al realizar los cálculos, por lo que a deformaciones se refiere las diagonales no trabajan. Las diagonales se encargan de los esfuerzos cortantes y se oponen a la deformación de la sección. Esta hipótesis se explica detalladamente en la sección “9 Cálculo de esfuerzos en cada tramo de torre, montantes y diagonales”.

A la hora de calcular el momento de inercia respecto al eje de giro, como estamos calculando las deformaciones axiales, no tenemos en cuenta las diagonales, y sólo nos fijamos en los montantes.

Para calcular el momento de inercia de un montante respecto al eje de giro sabemos que debemos de aplicar la ecuación:

$$I_x = I_{Mont} + A_{Mont} \cdot d^2$$

Siendo:

I_{Mont} Momento de inercia del montante respecto al eje de giro.

A_{Mont} Sección del montante.

d Distancia desde el centro de gravedad del montante al eje de giro.

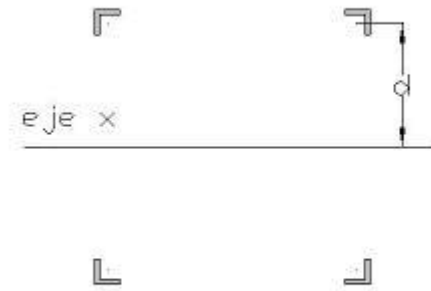
Para que los cálculos sean totalmente precisos debemos de tener en cuenta I_{Mont} , pero en la tabla de las propiedades de los tramos no tenemos este dato, con lo que si lo despreciamos, ocurre que el momento de inercia total de la sección será menor, con lo que al estar en el denominador en la ecuación:

$$\Delta_{B|A} = - \int_{x_A}^{x_B} \frac{M_f(x)}{EI_f} (x - x_A) dx$$

obtenemos unas deformaciones mayores. Con lo que al despreciar I_{Mont} de cada montante obtenemos un mayor rango de seguridad, al obtener unas deformaciones mayores, sin que

sean excesivas ya que el momento de inercia respecto al eje de giro de cada montante frente al producto del área y distancia es muy pequeño.

Siguiendo con los cálculos y observando las siguientes figuras, obtenemos primero el momento de inercia para el Caso 1 y luego para el Caso 2.



$$I_x = A_{Mont1} \cdot d_1^2 + A_{Mont2} \cdot d_2^2 + A_{Mont3} \cdot d_1^2 + A_{Mont4} \cdot d_4^2$$

Como la sección de los cuatro montantes y la distancia de sus respectivos centro de gravedad al eje de giro es la misma tendremos que.

$$I_x = 4 \cdot A_{Mont} \cdot d^2$$

Siendo:

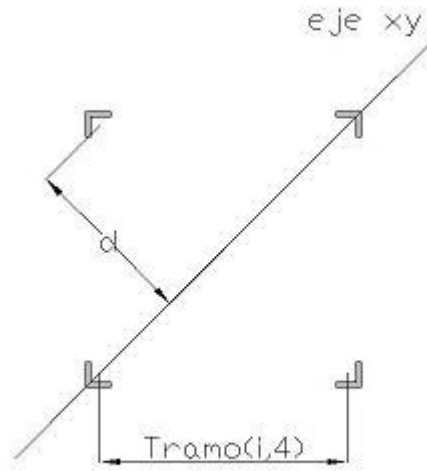
$A_{Mont} = Tramo(i, 6)$ Sección del montante

$$d = \frac{Tramo(i, 4)}{2}$$

$$I_x = 4 \cdot Tramo(i, 6) \cdot \left(\frac{Tramo(i, 4)}{2} \right)^2 = 4 \cdot Tramo(i, 6) \cdot \frac{Tramo(i, 4)^2}{4}$$

$$I_x = Tramo(i, 6) \cdot Tramo(i, 4)^2$$

Para el Caso 2 el cálculo es distinto, ya que la distancia del centro de gravedad de cada montante respecto al eje de giro es diferente. Si nos fijamos en la siguiente figura tenemos que:



$$d = \sqrt{\left(\frac{\text{Tramo}(i,4)}{2}\right)^2 + \left(\frac{\text{Tramo}(i,4)}{2}\right)^2} = \sqrt{\frac{\text{Tramo}(i,4)^2}{4} + \frac{\text{Tramo}(i,4)^2}{4}}$$

$$d = \sqrt{\frac{2 \cdot \text{Tramo}(i,4)^2}{4}} = \frac{\text{Tramo}(i,4)}{\sqrt{2}}$$

$$d^2 = \frac{\text{Tramo}(i,4)^2}{2}$$

En este Caso 2 tenemos dos montantes el 2 y 3 que su distancia desde su centro de gravedad al eje de giro es nulo con lo que tendremos que el momento de inercia total es:

$$I_{xy} = A_{Mont1} \cdot d_1^2 + A_{Mont4} \cdot d_4^2$$

Como la sección de los montantes 1 y 4 y la distancia de sus respectivos centro de gravedad al eje de giro es la misma tenemos que:

$$I_{xy} = 2 \cdot A_{Mont} \cdot d^2$$

Sustituyendo el valor de A_{Mont} y de d tenemos:

$$I_{xy} = 2 \cdot \text{Tramo}(i,6) \cdot \frac{\text{Tramo}(i,4)^2}{2} = \text{Tramo}(i,6) \cdot \text{Tramo}(i,4)^2$$

Vemos como el valor del momento de inercia para el Caso 1 y Caso 2 es el mismo, con lo que no hace falta realizar dos cálculos. Esto simplifica los cálculos.

$$I_x = I_{xy} = \text{Tramo}(i, 6) \cdot \text{Tramo}(i, 4)^2$$

7.2 CÁLCULO DE LAS FLECHAS.

Una vez que hemos calculado el momento de inercia de cada sección debemos de aplicar la ecuación:

$$\Delta_{B|A} = - \int_{x_A}^{x_B} \frac{M_f(x)}{EI_f} (x - x_A) dx$$

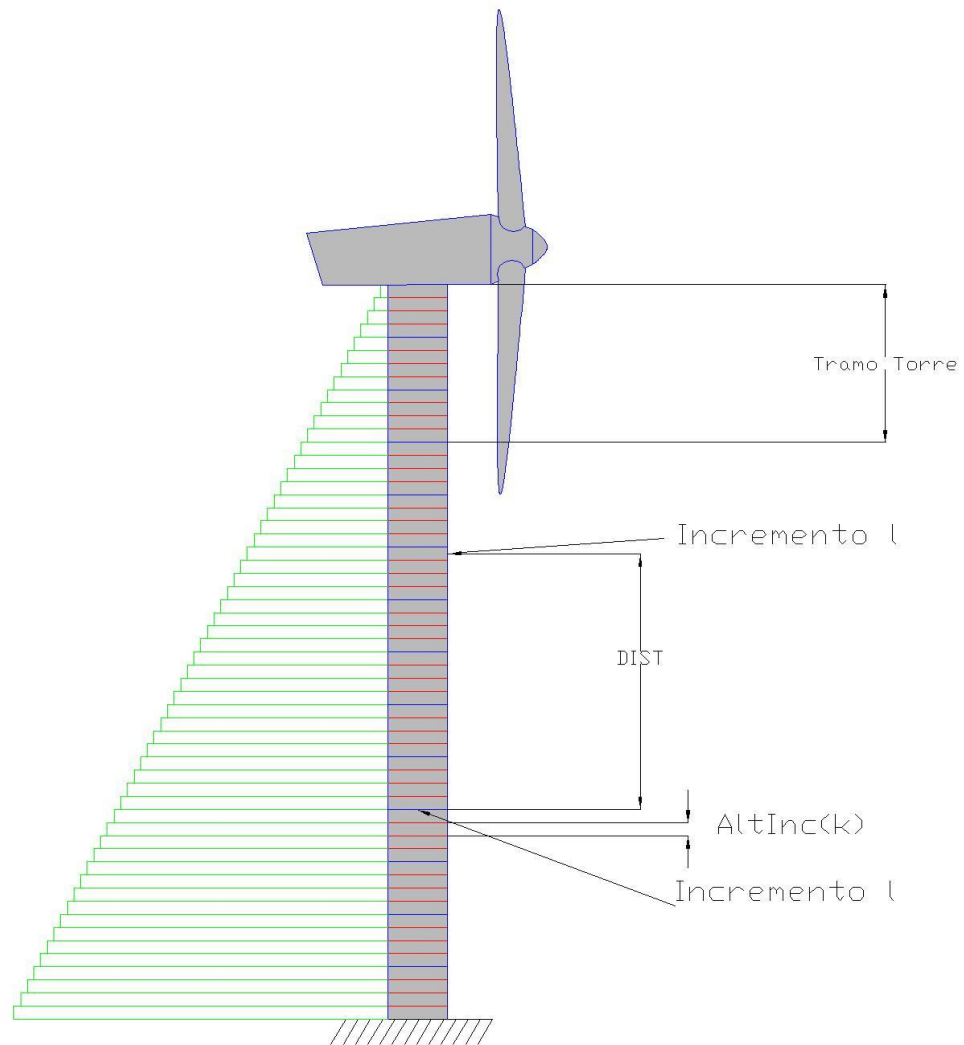
Como se ha comentado antes vemos que en cada incremento deberemos de multiplicar el área total del diagrama de momentos reducidos comprendida entre el incremento que se esté estudiando y el punto B por la distancia a su centro de gravedad.

Para encontrar el área de momento simplemente multiplicaremos el momento de cada incremento que hemos obtenido en el apartado “6 Cálculo de los momentos flectores, fuerzas cortantes y pesos”, por su respectiva altura de incremento. A continuación la multiplicamos por la distancia a B.

Esta distancia la guardaremos en una nueva variable que llamamos *DIST*. Esta distancia es la distancia entre el centro de cada incremento hasta el punto B.

Todo esto podemos verlo en la siguiente figura.

DIAGRAMA DE MOMENTOS DEBIDO A DISTINTAS SOLICITACIONES



Vemos como tenemos muchos factores para realizar los cálculos. De manera que inventamos una nueva variable llamada *Factor(100)* en la que guardamos el valor de cada producto de $E I_z$ de cada incremento, y además para obtener las deformaciones en m, cambiaremos todas las unidades.

$$Flecha = \frac{Momento \cdot AltInc \cdot DIST}{E \cdot I_z}$$

Si sustituimos las unidades:

$$Flecha = \frac{(N \cdot m) \cdot m \cdot m}{2,1 \cdot 10^6 \frac{kg}{cm^2} \cdot cm^2 \cdot m^2} = \frac{N \cdot m}{2,1 \cdot 10^6 kg} = \frac{N \cdot m}{2,1 \cdot 10^6 Kg} \cdot \frac{1Kg}{10N} = \frac{m}{2,1 \cdot 10^7}$$

En el variable $Factor(k)$ recordando la ecuación del momento de inercia:

$$I_x = I_{xy} = Tramo(i, 6) \cdot Tramo(i, 4)^2$$

Tendremos:

$$Factor(k) = Tramo(i, 6) \cdot Tramo(i, 4)^2 \cdot 21000000$$

Calculamos las flechas para el Caso 1 y el Caso 2, pero sólo pondremos el ejemplo del Caso1. El Caso 2 es exactamente igual, sólo que con sus respectivos valores.

En el apartado “6 Cálculo de los momentos flectores, fuerzas cortantes y pesos” hemos visto como en cada incremento tenemos 4 tipos de momentos, debido a las distintas solicitaciones. Estos cuatro momentos generan otras tantas flechas de deformación en cada incremento. Nosotros calculamos cada flecha debida a cada solicitación. Nueva variable $FlechaCaso1(100,4)$. 100 espacios para cada incremento, y 4 para cada distinta solicitación.

Tendremos pues:

$$FlechaCaso1(i, j) = \sum \frac{MomentosCaso1(k, j) \cdot DIST \cdot AltInc(k)}{Factor(k)}$$

Para obtener el sumatorio de cada incremento, en el programa nos valdremos de un bucle *For Next*.

7.3 EXPLICACIÓN RUTINA.

Declaración de variables.

```
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim l As Integer
Dim m As Integer
Dim Factor(100) As Double 'En esta variable guardaremos el valor de EIz de cada incremento, con el cambio de unidades
Dim DIST As Double
```

Calculamos el $Factor(k)$ de cada incremento. Para ello utilizamos dos bucles *For*. Uno dentro de otro. El primero indica que nos movemos desde el primer tramo hasta el último tramo de torre. Con el segundo nos movemos desde el primer incremento de cada tramo hasta el último. Para indicar en qué incremento estamos utilizamos la variable k . Al final de los dos bucles a esta variable k le sumaremos la unidad y volvemos a realizar los cálculos para el siguiente incremento. Al comienzo de la rutina damos a la variable k el valor de la unidad para que no haya problemas.

```
'Cálculo de las flechas producidas por el viento sobre la torre.

'Aplicaremos el segundo teorema de Moore para hallar las flechas.
'Flecha A-B=Integral(Mz/EIz)x dx
'Calcularemos el momento de inercia de cada incremento.

k = 1

For i = 1 To Dat(1)
  For j = 1 To NinC(1)

    Factor(k) = Tramo(i, 6) * Tramo(i, 4) ^ 2 * 2100000
    k = k + 1

  Next
Next
```

Aplicaremos ahora el segundo teorema de Mohr. Calculamos la flecha de cada incremento debidas a los diferentes momentos hallados en el apartado “6 Cálculo de los momentos flectores, fuerzas cortantes y pesos”. Como he comentado antes guardamos cada

flecha por separado. Por ejemplo la flecha por viento en la góndola del incremento i , para el Caso 1, queda almacenada en $FlechasCaso1(i,1)$.

Utilizamos tres bucles para hallar las flechas. El primero para hallar las flechas de los distintos momentos, desde el 1º hasta el 4º, una por cada distinto momento que tenemos. El segundo bucle, para calcular las flechas por un momento concreto desde el primer hasta el último incremento. El último bucle es para hacer el sumatorio, una vez que estamos en un momento concreto y queremos calcular la flecha en un incremento hacemos un bucle desde ese incremento hasta el incremento final para el sumatorio que nos da la flecha.

Para calcular la distancia hemos dicho que utilizamos la variable *DIST*, esta variable la pondremos a cero cada vez que entremos en el segundo bucle, para que no haya problemas. Le sumaremos al principio y al final del tercer bucle la mitad de la altura del incremento, para poder calcular la distancia correctamente.

'Aplicación de Mohr

k=0

For j = 1 To 4

For i = 1 To NinC(2) 'Desde el primer hasta el último "incremento"

DIST = 0

For k = i To NinC(2)

DIST = DIST + (AltInc(k) / 2)

$FlechasCaso1(i, j) = FlechasCaso1(i, j) + MomentosCaso1(k, j) * DIST * AltInc(k) / Factor(k) * MomentoEsfuerzos * Dist * Alt / F$

$FlechasCaso2(i, j) = FlechasCaso2(i, j) + MomentosCaso2(k, j) * DIST * AltInc(k) / Factor(k) * MomentoEsfuerzos * Dist * Alt / F$

DIST = DIST + (AltInc(k) / 2)

Next

Next

Next

8 CÁLCULO DE LOS MOMENTOS SECUNDARIOS CAUSADOS POR LAS DISTINTAS SOLICITACIONES

Hemos calculado los momentos que se producen por las distintas causas así como las deformaciones que causan. Pero aún no se han tenido en cuenta los momentos secundarios. Los momentos secundarios son aquellos que se producen al deformarse la torre, con lo que los pesos de esta quedan descentrados y producen unos momentos que pueden ser de escasa importancia pero que se deben de tener en cuenta.

Calculamos los momentos secundarios que pueden aparecer debidos a las distintas solicitudes que tenemos, que han causado distintas flechas.

- | | |
|--|-------------------|
| - Viento en góndola. | FlechasCaso1(i,1) |
| - Viento en aspas. | FlechasCaso1(i,2) |
| - Momento del peso de la góndola y aspas | FlechasCaso1(i,3) |
| - Viento en tramos de torre | FlechasCaso1(i,4) |

Al calcular el momento secundario de un incremento, calcularemos el sumatorio de momentos producido por el peso de todos los incrementos que queden por encima de él con excentricidad igual a la diferencia de flechas entre los incrementos. Debemos de tener en cuenta también el peso de la góndola y aspas, ya que estos pesos también están descentrados.

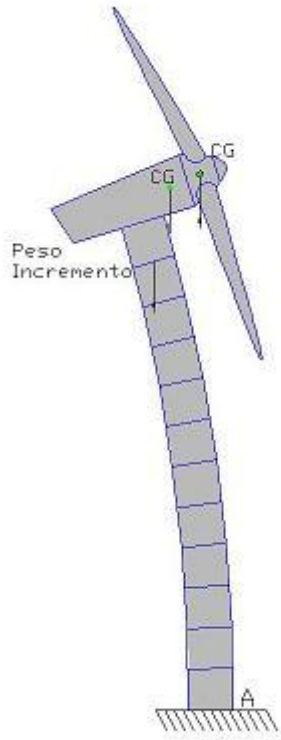
Tenemos una nueva variable, *MomentosSecunCaso1(100,4)*. Es una matriz con 100 espacios para cada incremento y 4 para la flecha causada por las distintas solicitudes mencionadas anteriormente.

Pondremos un ejemplo, para los momentos secundarios causados por el viento en la góndola en el Caso 1. Para el Caso 2 es exactamente igual.

Se calcularán los pesos individuales de cada incremento, guardándolos en la variable *PesoIncremento(100,4)*.

Primero calculamos el momento secundario causado por el peso de la góndola y aspas. Se debe de tener en cuenta que los pesos de estos elementos están descentrados con lo que el momento que puedan generar, es de signo contrario al que puedan generar los pesos propios de la torre. La distancia es la distancia de descentrado más la distancia de la flecha.

Pondremos el signo negativo. Multiplicamos por 10 para tener Newtons.



$$\begin{aligned} \text{MomentosSecunCaso1}(i, 1) = \\ (-(\text{Dat}(2) \cdot \text{Dat}(10)) \cdot (\text{Dat}(4) + \text{FlechasCaso1}(i, k)) - \text{Dat}(3) \cdot \text{Dat}(10)) \\ \cdot (\text{Dat}(5) + \text{FlechasCaso1}(i, k))) \cdot 10 \end{aligned}$$

A continuación le sumaremos el momento causado por lo pesos de la torre que están descentrados:

$$\begin{aligned} \text{MomentosSecunCaso1}(i, 1) = \\ \text{MomentosSecunCaso1}(i, 1) + (\text{PesoIncremento}(i) \cdot \text{Dat}(10)) \\ \cdot (\text{FlechasCaso1}(j, 1) - \text{FlechasCaso1}(i, 1))) \cdot 10 \end{aligned}$$

Por último en una nueva variable `MomentoSecundarioTotal(2)`, con dos espacios de memoria, un resultado para cada caso de estudio, conseguimos el valor del momento secundario total de la base. Simplemente sumaremos los momentos secundarios de las cuatro solicitaciones del último incremento de la torre.

$$\begin{aligned} & \text{MomentoSecundarioTotal}(2) \\ &= \text{MomentoSecunCaso1}(\text{Ninc}(2), 1) + \text{MomentoSecunCaso1}(\text{Ninc}(2), 2) \\ &- \text{MomentoSecunCaso1}(\text{Ninc}(2), 3) + \text{MomentoSecunCaso1}(\text{Ninc}(2), 4) \end{aligned}$$

8.1 EXPLICACIÓN RUTINA.

Declaración de variables.

```
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim PesoIncremento(100) As Double
```

Con dos bucles conseguimos el peso de cada incremento por separado.

```
k = 1
For i = 1 To Dat(1)
  For j = 1 To NInc(1)
    PesoIncremento(k) = Tramo(i, 5) / NInc(1)
    k = k + 1
  Next
Next
```

Utilizamos tres bucles para hallar los momentos secundarios. El primero para hallar los momentos secundarios, desde el 1º hasta el 4º, una por cada distinto momento que tenemos por cada solicitud antes comentada. El segundo bucle, para calcular los momentos desde el primer hasta el último incremento. El último bucle es para hacer el sumatorio, una vez que estamos en un incremento concreto y queremos calcular el momento secundario total debemos de tener en cuenta el peso de los incrementos que tiene por encima, ya que al estar descentrados producen un momento en ese incremento. De manera que hacemos un bucle desde el primer incremento hasta el incremento que se está estudiando.

k=0

For k = 1 To 4 '(Nº de "Momentos" distintos que tenemos en la base, por cada una de las flechas de la anterior lista)

For i = 1 To NinC(2)

MomentosSecunCaso1(i, k) = -(Dat(2) * Dat(10)) * (Dat(4) + FlechasCaso1(i, k)) - Dat(3) * Dat(10) * (Dat(5) + FlechasCaso1(i, k))

MomentosSecunCaso2(i, k) = -(Dat(2) * Dat(10)) * (Dat(4) + FlechasCaso2(i, k)) - Dat(3) * Dat(10) * (Dat(5) + FlechasCaso2(i, k))

'Con este bucle sumaremos a cada "incremento", el momento causado por el viento que esté por encima de cada "incremento"

For j = 1 To i

$$\text{MomentosSecunCaso1}(i, k) = \text{MomentosSecunCaso1}(i, k) + (\text{Peso}(j) * \text{Dat}(10) * (\text{FlechasCaso1}(j, k) - \text{FlechasCaso1}(i, k))) * 10$$

$$\text{MomentosSecunCaso2}(i, k) = \text{MomentosSecunCaso2}(i, k) + (\text{Peso}(j) * \text{Dat}(10) * (\text{FlechasCaso2}(j, k) - \text{FlechasCaso2}(i, k))) * 10$$

Next

Next

Next

$$\begin{aligned} \text{MomentoSecundarioTotal}(1) &= \text{MomentosSecunCaso1}(\text{NInc}(2), 1) + \text{MomentosSecunCaso1}(\text{NInc}(2), 2) - \\ &\text{MomentosSecunCaso1}(\text{NInc}(2), 3) + \text{MomentosSecunCaso1}(\text{NInc}(2), 4) \end{aligned}$$

$$\begin{aligned} \text{MomentoSecundarioTotal}(2) &= \text{MomentosSecunCaso2}(\text{NInc}(2), 1) + \text{MomentosSecunCaso2}(\text{NInc}(2), 2) - \\ &\text{MomentosSecunCaso2}(\text{NInc}(2), 3) + \text{MomentosSecunCaso2}(\text{NInc}(2), 4) \end{aligned}$$

9 CÁLCULO DE ESFUERZOS EN CADA TRAMO DE TORRE, MONTATES Y DIAGONALES.

Primeramente realizaremos un estudio general de cómo se comporta la torre ante las distintas solicitaciones que podemos encontrar. Tanto para el Caso 1 como para el Caso 2. Dependiendo a que reacciones sometamos la estructura, las diagonales de los tramos Tipo 1 y Tipo 2, trabajan a distintos esfuerzos. Los calcularemos detalladamente mostrando imágenes para su fácil comprensión.

9.1 FLEXIÓN PURA EN DIRECCIÓN TRANSVERSAL A LA SECCIÓN. CASO 1

Def. Flexión Pura. La flexión pura como definición es aquella en que los esfuerzos cortantes son igual a cero. En nuestro estudio tomaremos que esta flexión es producida por el momento producido por los pesos descentrados de la góndola y el peso de las aspas. Además de los momentos producidos por los pesos de cada incremento descentrados a causa de la deformación de la torre.

Al no haber esfuerzos cortantes las diagonales no intervienen en el cálculo y los únicos elementos que se deforman y en consecuencia que sufren son los montantes.

Los tramos están sometidos a dos momentos flectores de dirección contraria, lo que se traduce en cuatro fuerzas iguales, una en cada montante. Todas estas fuerzas están a la misma distancia y tienen el mismo módulo, unas a tracción y otras a compresión. Así que el momento respecto al eje transversal de la sección que producen será igual al momento al que está sometido el tramo:

$$4 \cdot F_{Mon} \cdot d = M_{Flec}$$

Obtendremos las fuerzas en los montantes despejando la ecuación y sustituyendo el valor de la distancia d .



$$d = \frac{Tramo(i,4)}{2}$$

$$F_{Mon} = \frac{M_{Flec}}{2 \cdot Tramo(i,4)}$$

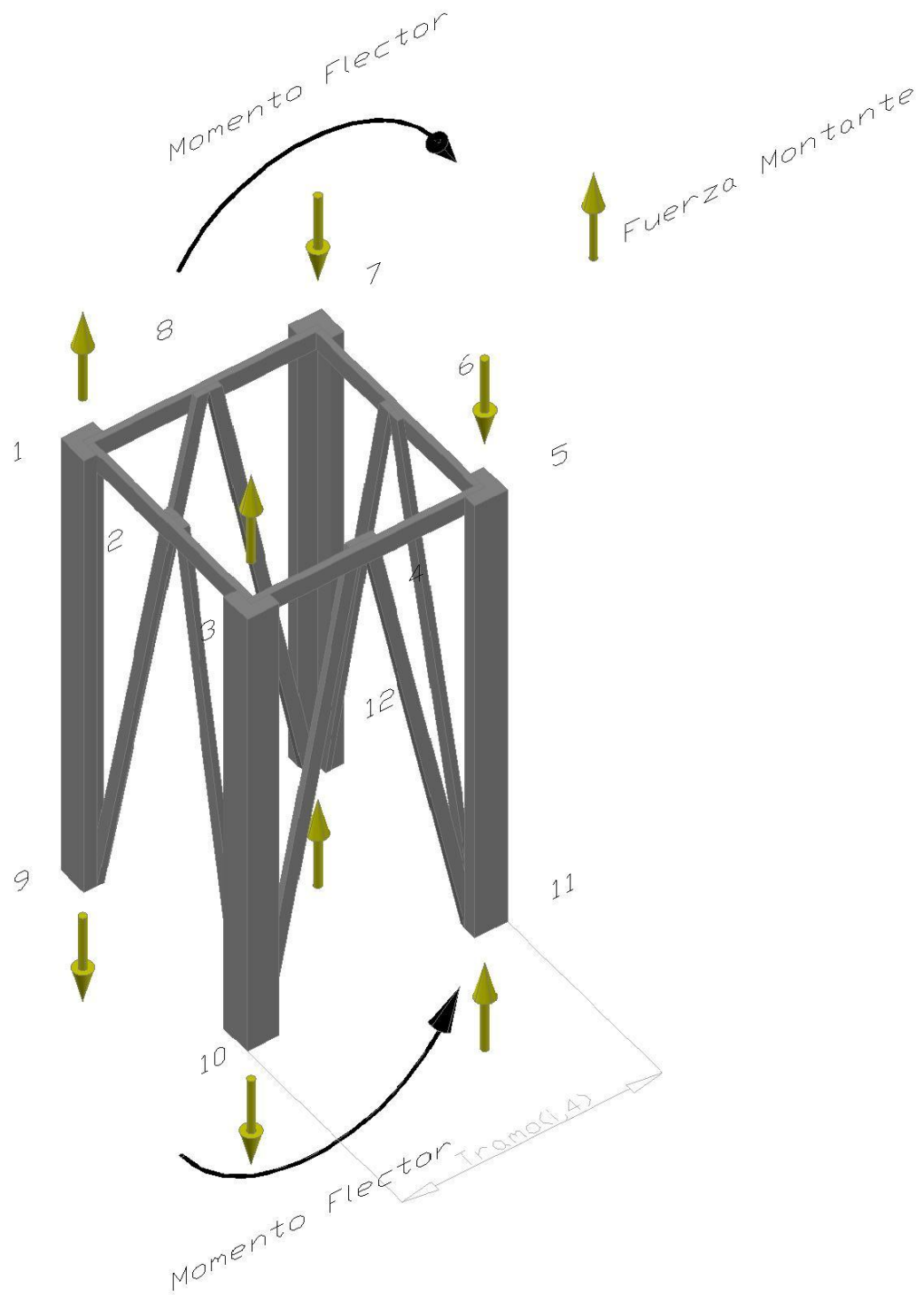
$Tramo(i,4)$ = Lado del Tramo

Conseguiremos el esfuerzo en el montante simplemente dividiendo la fuerza en el montante por la sección del mismo

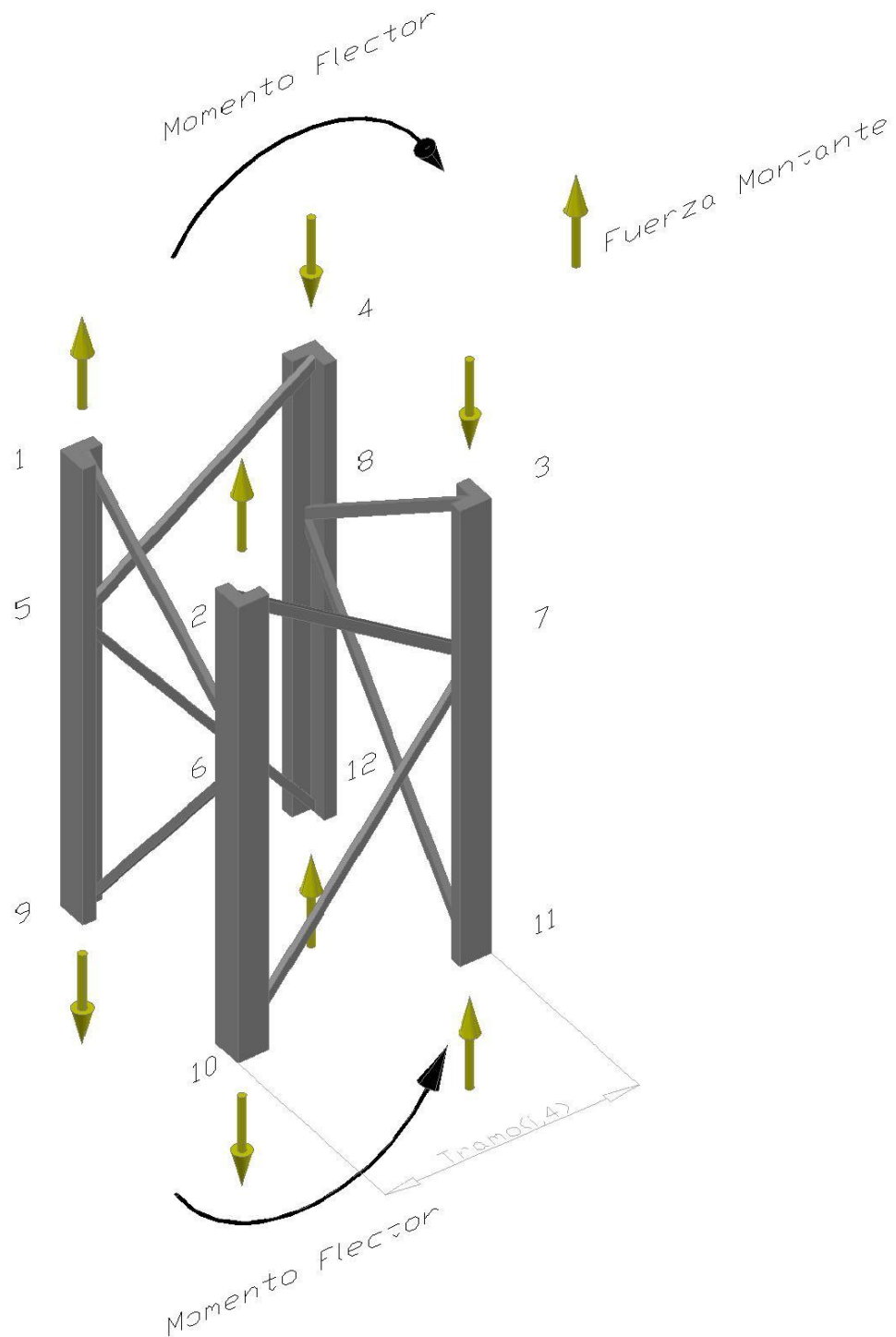
$$Es_{Mon} = \frac{M_{Flec}}{2 \cdot Tramo(i,4) \cdot Tramo(i,6)}$$

$Tramo(i,6)$ = Sección del Montante

TRAMO TIPO 1



TRAMO TIPO 2



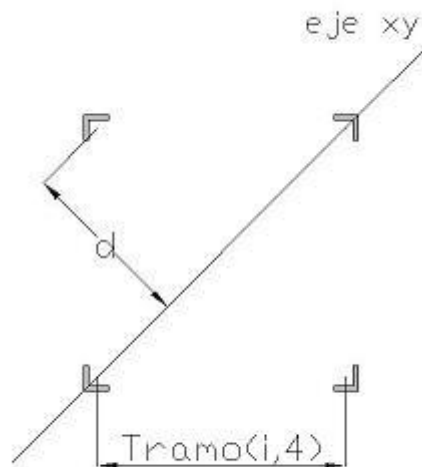
9.2 FLEXIÓN PURA EN DIRECCIÓN DIAGONAL A LA SECCIÓN. CASO 2

El Caso 2 es muy parecido al anterior. La única diferencia es que la dirección del momento en este caso al ser distinta, el momento es de giro alrededor del eje diagonal de la sección, por lo que tomamos momentos respecto a este eje. Al igual que en el Caso 1 los momentos se traducen en fuerzas de tracción y compresión en los montantes para que se dé el equilibrio. Pero en este caso el momento como he dicho antes es de giro alrededor del eje diagonal de la sección, con lo que pasa por dos montantes. Con lo cual estos dos montantes producen un momento nulo, es decir están descargados. Solamente dos montantes están sometidos a fuerzas de igual módulo, un montante está a sometido a un esfuerzo de tracción y el otro a compresión.

El momento que producen estas dos fuerzas es igual al momento al que está sometido el tramo, todo ello para que se dé el equilibrio.

$$2 \cdot F_{Mon} \cdot d = M_{Flec}$$

Obtendremos las fuerzas en los montantes despejando la ecuación y sustituyendo el valor de la distancia d . Con la siguiente figura obtenemos y verificamos rápidamente el valor de d .



$$d = \sqrt{\left(\frac{\text{Tramo}(i, 4)}{2}\right)^2 + \left(\frac{\text{Tramo}(i, 4)}{2}\right)^2} = \sqrt{\frac{\text{Tramo}(i, 4)^2}{4} + \frac{\text{Tramo}(i, 4)^2}{4}}$$

$$d = \sqrt{\frac{2 \cdot \text{Tramo}(i, 4)^2}{4}} = \frac{\text{Tramo}(i, 4)}{\sqrt{2}}$$

$$d = \frac{\text{Tramo}(i, 4)}{\sqrt{2}}$$

$$F_{Mon} = \frac{M_{Flec}}{\sqrt{2} \cdot \text{Tramo}(i, 4)}$$

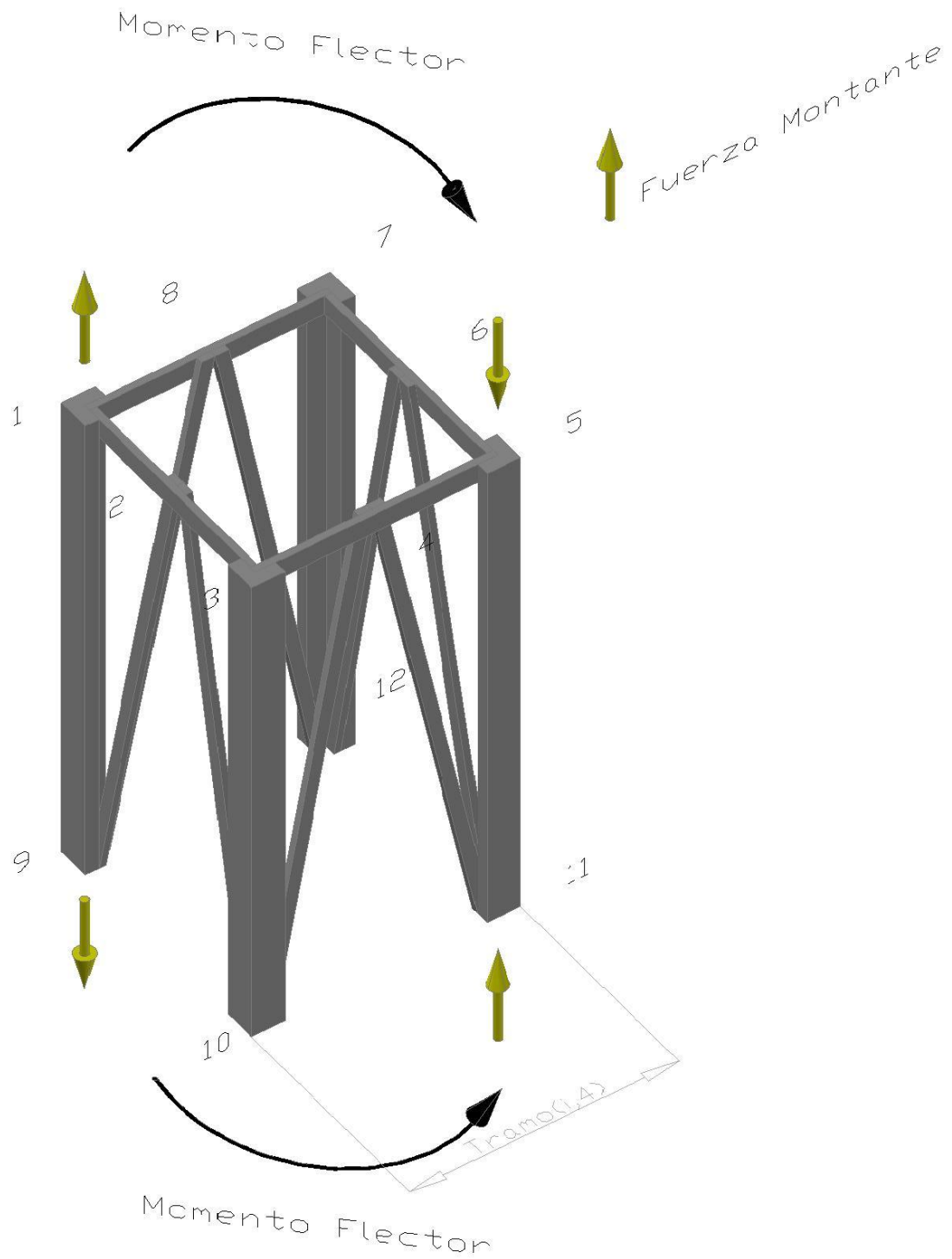
$\text{Tramo}(i, 4)$ = Lado del Tramo

Conseguiremos el esfuerzo en el montante simplemente dividiendo la fuerza en el montante por la sección del mismo

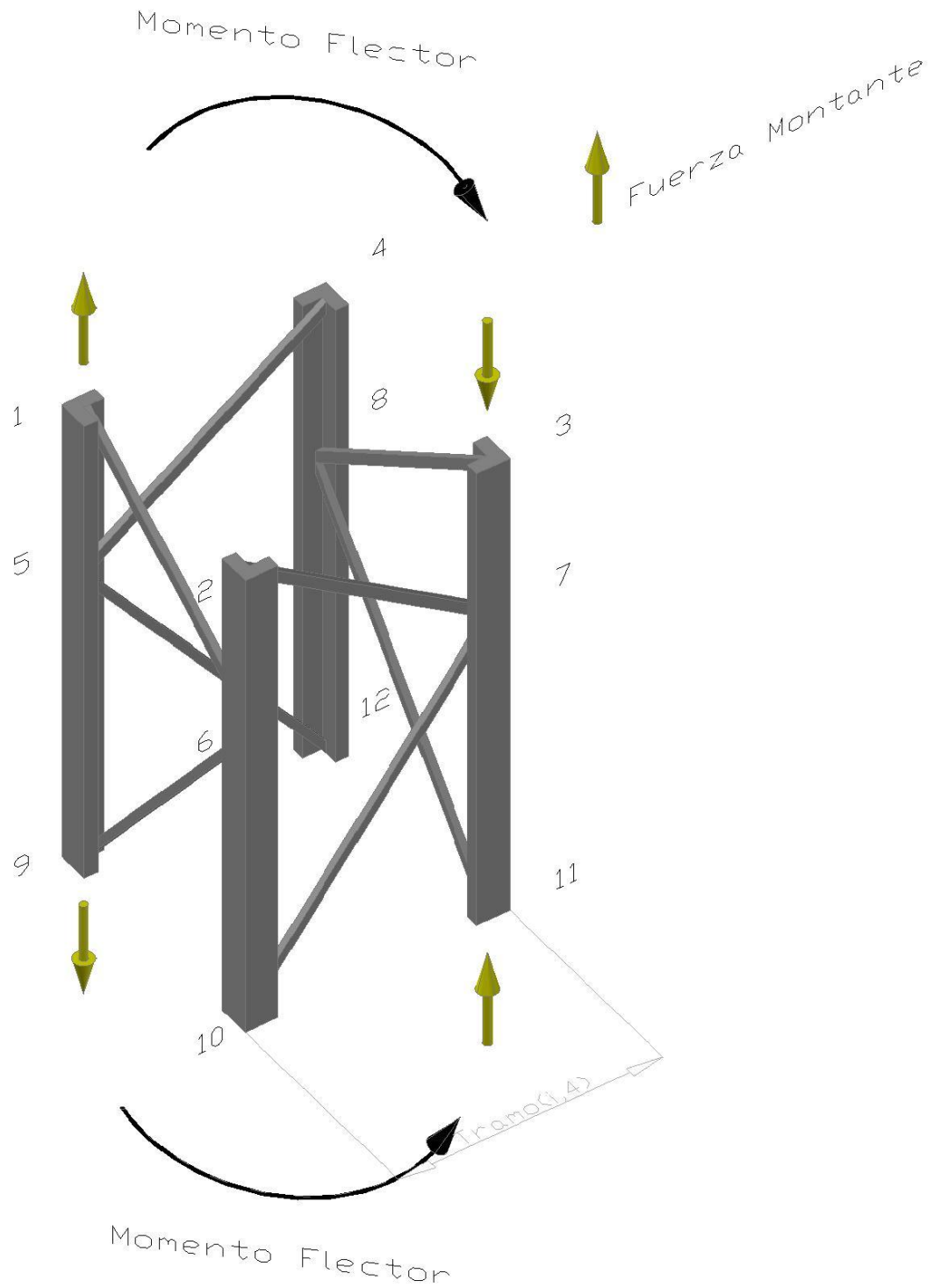
$$ES_{Mon} = \frac{M_{Flec}}{\sqrt{2} \cdot \text{Tramo}(i, 4) \cdot \text{Tramo}(i, 6)}$$

$\text{Tramo}(i, 6)$ = Sección del Montante

TRAMO TIPO 1

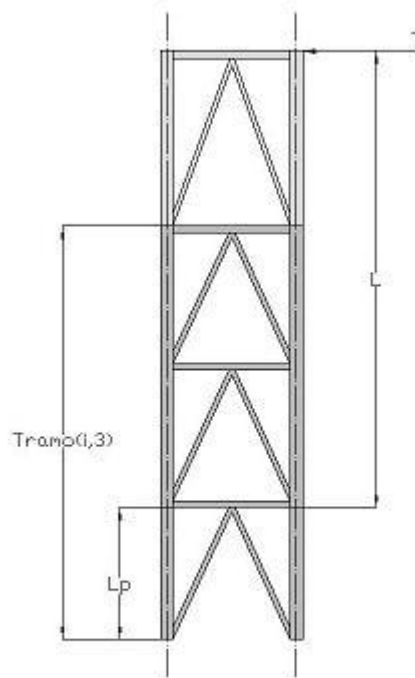


TRAMO TIPO 2



9.3 MOMENTOS FLECTORES CON FUERZAS CORTANTES EN DIRECCIÓN TRANSVERSAL A LA SECCIÓN. CASO 1

Estudiaremos a continuación los momentos más frecuentes, que son debidos a las fuerzas del viento. Supondremos que tenemos una fuerza T y que hacemos el estudio de un paso de tramo cuya sección superior está a una distancia L y la distancia a la sección inferior es $L + L_p$. De manera que el momento flector en la parte superior e inferior son respectivamente:



$$\text{Momento parte Superior} = T \cdot L$$

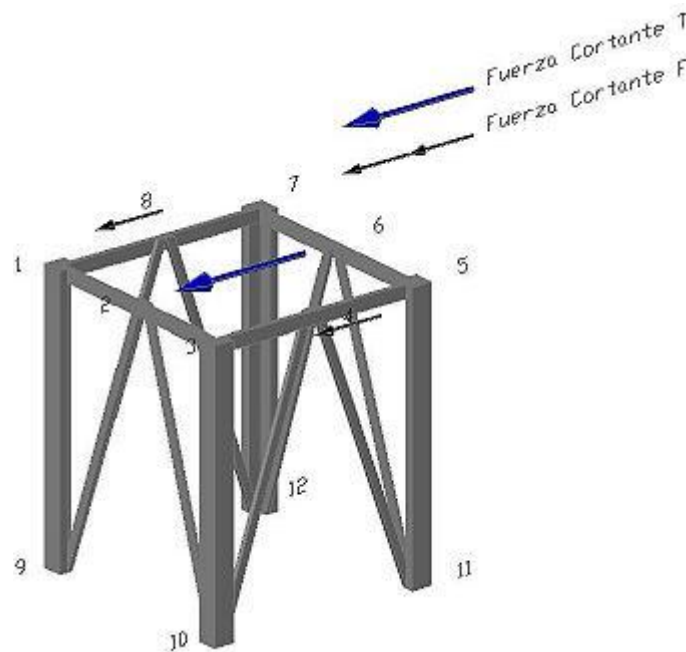
$$\text{Momento parte Inferior} = T \cdot (L + L_p)$$

En este apartado deberemos de distinguir los esfuerzos en las diagonales para el tramo Tipo 1 y tramo Tipo 2.

9.3.1 Esfuerzos en tramo Tipo 1.

Las diagonales de un tramo o de un paso de tramo sólo trabajan a tracción o compresión, por lo que en este caso como tenemos esfuerzos cortantes las descompondremos en fuerzas paralelas a las caras del tramo. En el Caso 1 la fuerza cortante T tiene la misma dirección que dos caras opuestas del paso, con lo que las diagonales de estas dos caras serán las encargadas de absorber esta fuerza. Las fuerzas en las diagonales producirán fuerzas en los montantes para que el sumatorio de fuerzas sea igual a cero.

Descompones las fuerzas cortantes T . Cada cara estará sometida a una fuerza igual a la mitad de T . A partir de ahora utilizaremos la variable F para definir esta fuerza.



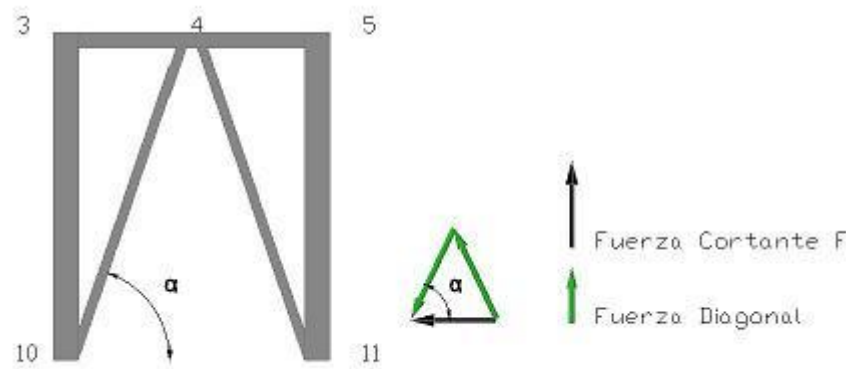
Empezando por la parte superior del paso, las fuerzas F las aplicamos en los 4 y 8. Nudos de unión entre diagonales superiores e inferiores. En los nudos 2 y 6 que son del mismo tipo no hay fuerzas, ya que no habrá fuerza cortante en ese eje, es decir en la dirección de esa cara.

Al aplicar la fuerza en el nudo 4, tenemos que. La diagonal superior esta cargada con la fuerza cortante $F/2$ aplicada en cada extremo, y una fuerza F en su centro de

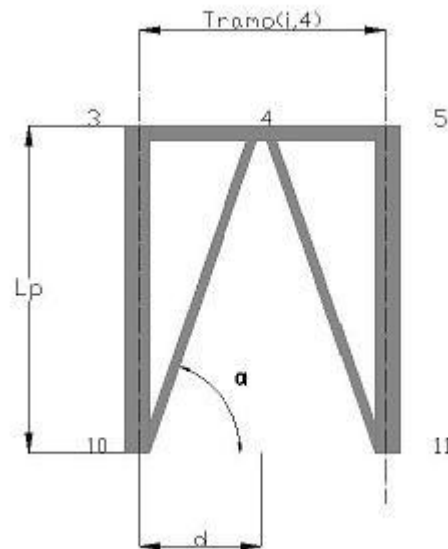
sentido contrario, por lo que la media diagonal superior trabaja a compresión y la otra media a tracción.

Si aplicamos equilibrio de fuerzas en este nudo, teniendo en cuenta que toda la diagonal superior es un elemento único y que toda ella absorbe la fuerza F podremos obtener las fuerzas de las diagonales.

La suma de fuerzas en estos nudos las vemos en la siguiente figura. Estas fuerzas dependen del ángulo α , el cual depende de la anchura y altura del paso. Con este ángulo y con la fuerza F conseguimos hallar las fuerzas en las diagonales 4-10, 4-11, 8-9 y 8-12.



Las diagonales inferiores de cada paso tendrán una fuerza de mismo módulo, para que se dé el equilibrio en el eje Y , y además una será a tracción y otra a compresión para que en el eje X se sumen y sean iguales a F . De manera que sabemos que la componente horizontal de estas diagonales es igual a $F/2$. Con todo ello y con las siguientes figuras obtenemos el valor de las fuerzas en las diagonales. Primero debemos de obtener el ángulo al que están las diagonales inferiores.



$$\text{tag}(\alpha) = \frac{L_p}{d/2}$$

Sustituimos L_p y d por sus respectivos valores

$$L_p = \frac{\text{Altura Tramo}}{N^{\circ} \text{ Pasos}} = \frac{\text{Tramo}(i,3)}{\text{Tramo}(i,2)}$$

$$d = \text{Lado del Tramo} = \text{Tramo}(i,4)$$

$$\text{tag}(\alpha) = \frac{2 \cdot \text{Tramo}(i,3)}{\text{Tramo}(i,2) \cdot \text{Tramo}(i,4)}$$

$$\alpha = \arctg\left(\frac{2 \cdot \text{Tramo}(i,3)}{\text{Tramo}(i,2) \cdot \text{Tramo}(i,4)}\right)$$

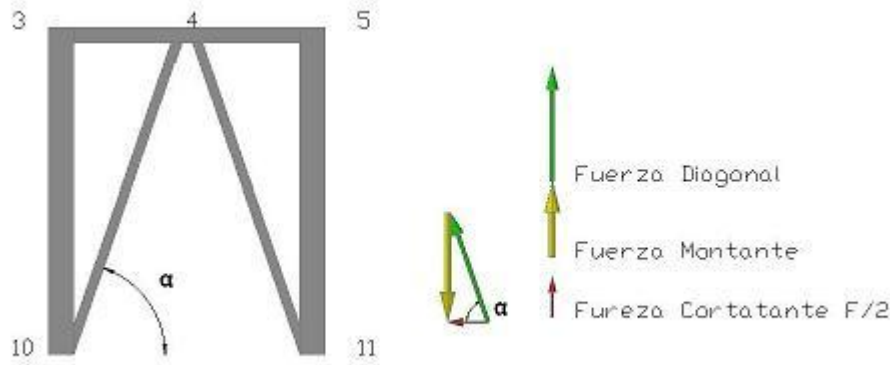
La suma de las componentes horizontales es igual a F

$$F = 2 \cdot F_{\text{Dialnf}} \cdot \cos(\alpha)$$

$$F_{\text{Dialnf}} = \frac{F}{2 \cdot \cos(\alpha)}$$

El resultado es que tenemos las 3 diagonales cargadas, dos a compresión y una a tracción. Cada una de las diagonales inferiores va a un nudo diferente, que está en un montante diferente. Como consecuencia tendremos que los cuatro montantes estarán

cargados también. Estas fuerzas las denominaremos como en el caso anterior por F_{Mon} . Dos están a compresión y otras dos a tracción. La fuerza de la diagonal y la de montante en dirección vertical dan como resultado otra en dirección horizontal de la fuerza cortante que actúa en la parte inferior del paso de tramo.



Obtenemos tanto la fuerza en el montante como la fuerza horizontal que aparece en la parte inferior del tramo.

$$FuerzaHorizontal = F_{Dialnf} \cdot \cos(\alpha) = \frac{F}{2 \cdot \cos(\alpha)} \cdot \cos(\alpha) = \frac{F}{2}$$

Esta cuatro fuerzas horizontales nos da como resultado una fuerza T en la parte inferior del tramo.

Sigamos calculando la fuerza de los montantes.

$$F_{Mon} = F_{Dialnf} \cdot \sin(\alpha)$$

$$F_{Mon} = \frac{F \cdot \cos(\alpha)}{2 \cdot \sin(\alpha)} = \frac{F \cdot \tan(\alpha)}{2}$$

Sustituyendo los valores de F y $\tan(\alpha)$

$$F_{Mon} = \frac{1}{2} \cdot \frac{T}{2} \cdot \frac{2 \cdot Tramo(i,3)}{Tramo(i,2) \cdot Tramo(i,4)} = \frac{T \cdot Tramo(i,3)}{2 \cdot Tramo(i,2) \cdot Tramo(i,4)}$$

Recordamos que ya hemos obtenido la fuerza en las diagonales inferiores, y sustituyendo el valor de F tenemos que

$$F_{DiaInf} = \frac{F}{2 \cdot \cos(\alpha)} = \frac{1}{2 \cdot \cos(\alpha)} \cdot \frac{T}{2} = \frac{T}{4 \cdot \cos(\alpha)}$$

Sabemos la fuerza en cada barra, de manera que hayamos los esfuerzos.

En los montantes los esfuerzos son iguales a pesar de que unos están a tracción y los otros a compresión.

$$ES_{Mon} = \frac{F_{Mon}}{Tramo(i, 6)} = \frac{T \cdot Tramo(i, 3)}{2 \cdot Tramo(i, 2) \cdot Tramo(i, 4) \cdot Tramo(i, 6)}$$

$Tramo(i, 6)$ = Sección del Montante

Con las diagonales inferiores ocurre lo mismo, todas tienen la misma fuerza sólo que unas están a tracción y otras a compresión.

$$ES_{DiaInf} = \frac{F_{DiaInf}}{Tramo(i, 8)} = \frac{T}{4 \cdot \cos(\alpha) \cdot Tramo(i, 8)}$$

$Tramo(i, 8)$ = Sección de la Diagonal Inferior

Por último tenemos las diagonales superiores. Como he comentado antes es un único elemento de manera que absorbe todo el esfuerzo cortante, con lo que el esfuerzo en la diagonal superior, sustituyendo el valor de F será:

$$ES_{DiaSup} = \frac{F}{2 \cdot Tramo(i, 7)} = \frac{T}{4 \cdot Tramo(i, 7)}$$

$Tramo(i, 7)$ = Sección de la Diagonal Superior

Hasta ahora simplemente hemos realizado un estudio de fuerzas de equilibrio en los nudos, sin tener en cuenta para nada el momento que generan estos esfuerzos cortantes. Pero como hemos comentado antes, con el estudio de flexión pura, las fuerzas que contrarrestan estos momentos son las fuerzas de los montantes. De manera que hay que comprobar que los momentos producidos por las fuerzas de los montantes son los que produce la fuerza cortante.

El momento que producen las fuerzas en los montantes alrededor del eje transversal de giro es:

$$M = 4 \cdot F_{Mon} \cdot d$$

$$d = \frac{Tramo(i, 4)}{2}$$

$Tramo(i, 4) = \text{Lado del Tramo}$

Si sustituimos el valor de la fuerza del montante, F_{Mon} , que hemos obtenido antes tendremos que:

$$M = 4 \cdot \frac{T \cdot Tramo(i, 3)}{2 \cdot Tramo(i, 2) \cdot Tramo(i, 4)} \cdot \frac{Tramo(i, 4)}{2} = \frac{T \cdot Tramo(i, 3)}{Tramo(i, 2)}$$

Teniendo en cuenta que:

$$\frac{Tramo(i, 3)}{Tramo(i, 2)} = L_p$$

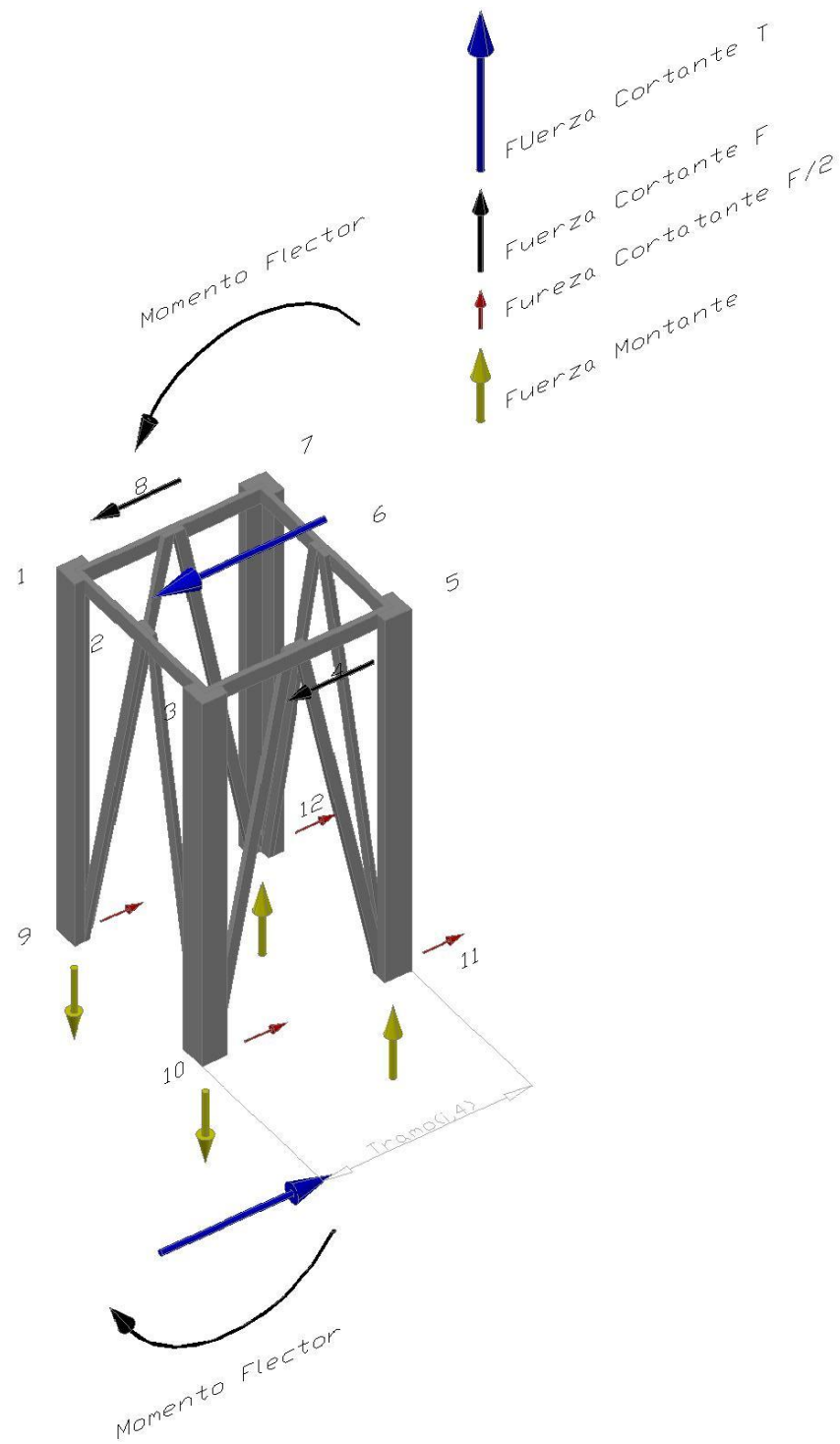
Tendremos que:

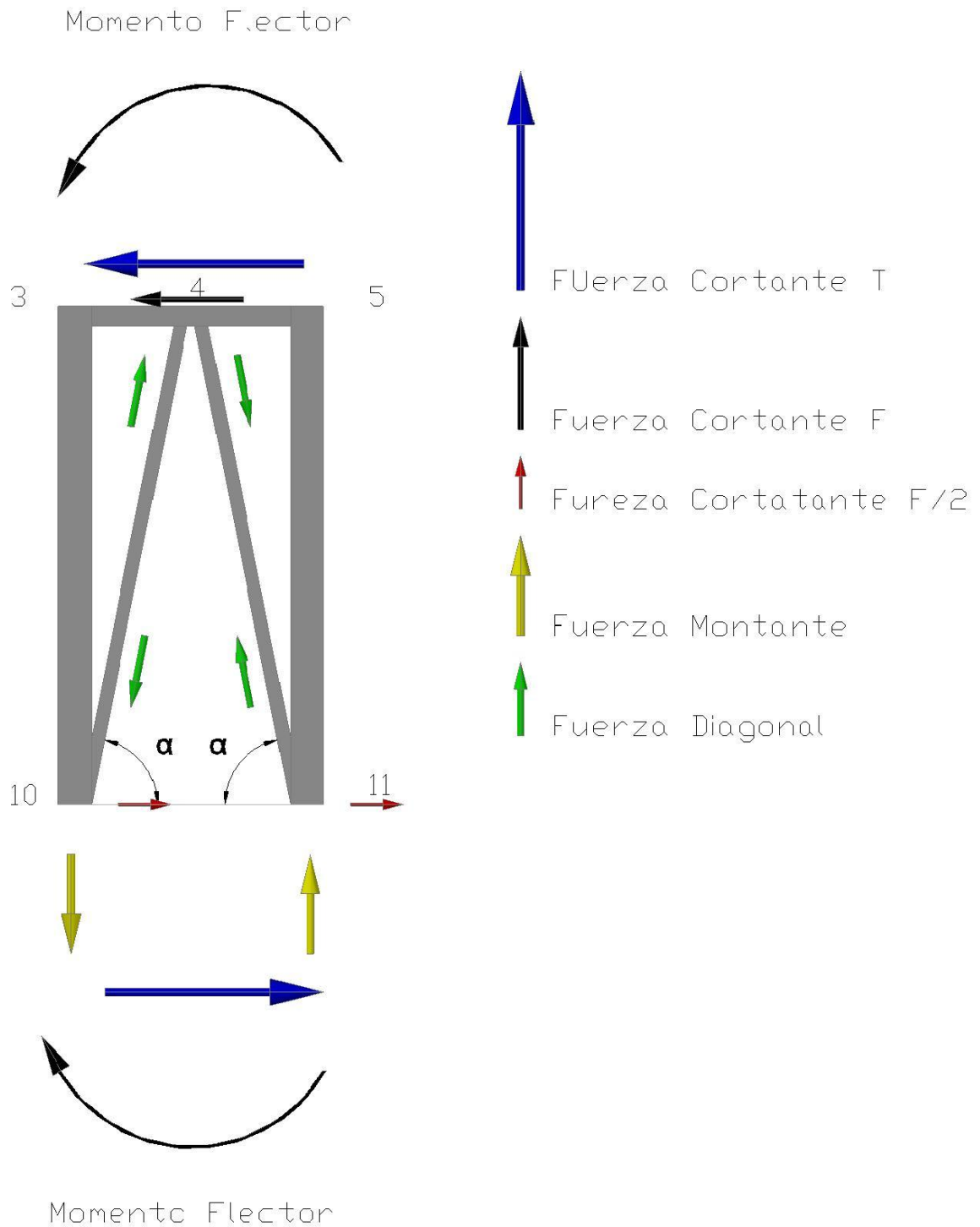
$$M = T \cdot L_p$$

Así pues que las fuerzas de los montantes producirán un momento igual al esfuerzo cortante por la diferencia de alturas entre la sección superior e inferior de cada paso de tramo. Como hemos partido de la teoría que los esfuerzos cortantes son absorbidos por las diagonales, las fueras de los montantes serán las encargadas de contrarrestar los momentos, otra de las teorías que hemos enunciado. Los tramos que se encuentran por encima del estudiado transmiten a los montantes unas fuerzas que producen el momento en la parte superior e inferior respectivamente

$$\begin{aligned} \text{Momento parte Superior} &= T \cdot L \\ \text{Momento parte Inferior} &= T \cdot (L + L_p) \end{aligned}$$

Idea que se ha comentado al principio del estudio.





Con la ayuda de la figura podemos ver el resultado de cómo trabajan las diagonales es el siguiente:

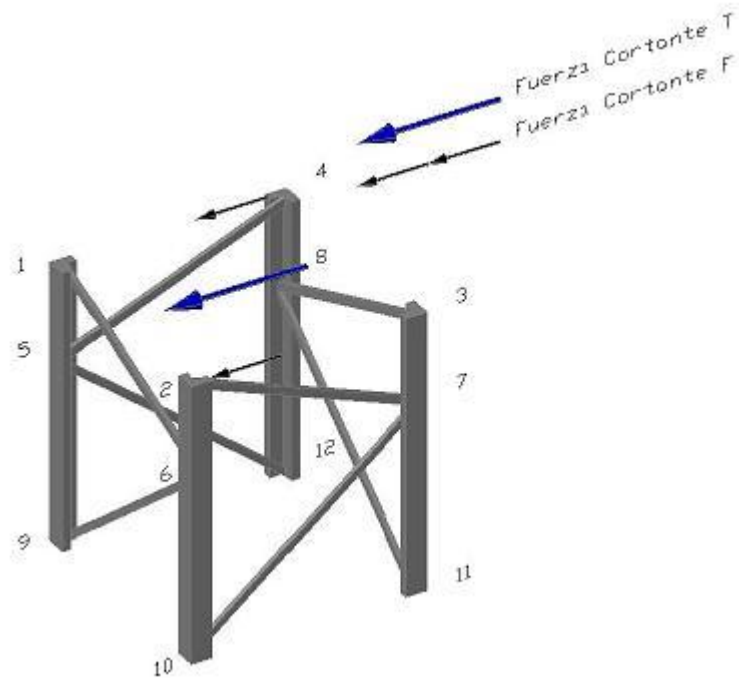
- Montante 1-9 a compresión.
- Montante 3-10 a compresión.
- Montante 7-12 a tracción.
- Montante 5-11 a tracción.
- Diagonal Inferior 2-9 y 2-10 descargada.
- Diagonal Inferior 6-11 y 6-12 descargada.
- Diagonal Inferior 4-10 a compresión.
- Diagonal Inferior 4-11 a tracción.
- Diagonal Inferior 8-9 a compresión.
- Diagonal Inferior 8-12 a tracción.

9.3.2 Esfuerzos en tramo Tipo 2

En un principio las bases que aplicaremos para el cálculo de las diagonales son iguales. Comenzaremos con los esfuerzos en las diagonales fruto de las fuerzas cortantes debidas al viento dentro del tramo. Las fuerzas en las diagonales producirán fuerzas en los montantes para que el sumatorio de fuerzas sea igual a cero.

Como en los tramo de Tipo 1, las diagonales de un tramo, solamente van a trabajar a tracción o a compresión. De igual manera que hemos hecho con el tramo Tipo 1 descomponemos la fuerza cortante T en fuerzas paralelas a la cara del tramo. Estas fuerzas en el Caso 1 tienen igual dirección que dos caras opuestas del paso. Las diagonales de estas dos caras serán las encargadas de absorber esta fuerza.

Descomponemos T .



En este Caso 1 las caras del tramo que trabajan son, 2,3,10 y 11. 1,4,9 y 12. Con lo que sólo estudiaremos esas caras. Las los otras dos caras no trabajan.

Esta fuerza F es el resultado de la suma de dos fuerzas en los nudos 2, 4, 10 y 12. En estos nudos se unen las diagonales y montantes que van a trabajar:

$$F = F_{Mont} + F_{Dia}$$

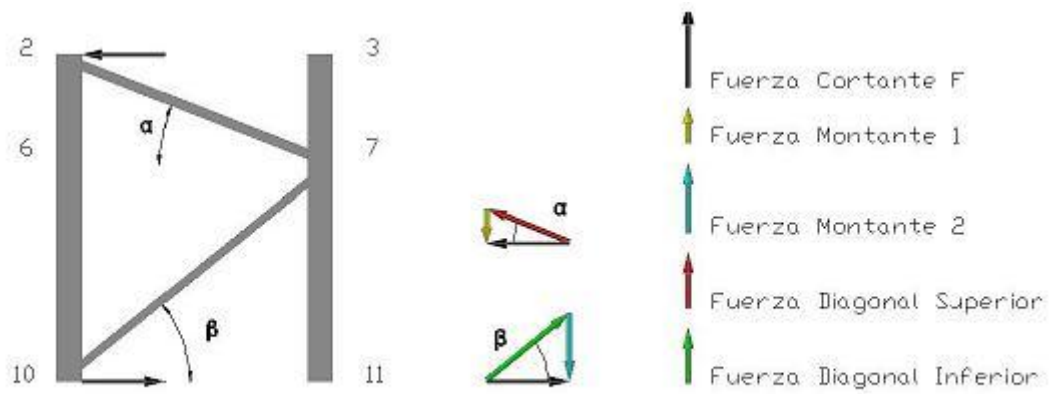
En la diagonal superior tendremos:

$$F = F_{Mont1} + F_{DiaSup}$$

En la diagonal inferior tendremos:

$$F = F_{Mont2} + F_{DiaInf}$$

Como resultado de las sumas vectoriales que se dan en estos nudos podemos despejar, F_{Mont1} , F_{DiaSup} , F_{Mont2} , F_{DiaInf}



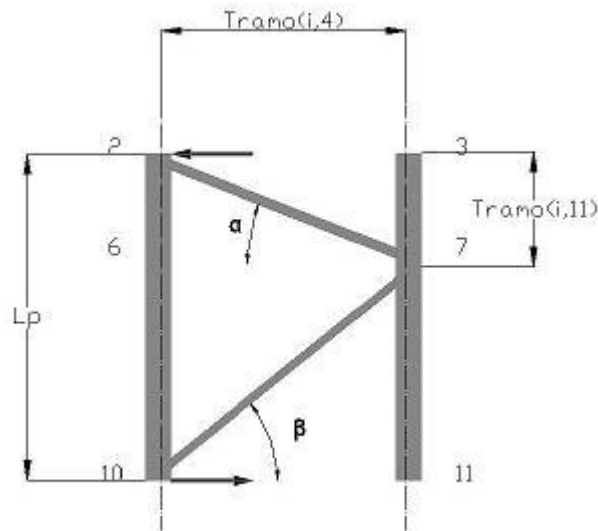
$$F_{Mont1} = F \cdot \tan \alpha$$

$$F_{DiaSup} = \frac{F}{\cos(\alpha)}$$

$$F_{Mont2} = F \cdot \tan \beta$$

$$F_{DiaInf} = \frac{F}{\cos(\beta)}$$

Con la siguiente figura obtenemos el valor de los ángulo α y β .



$$\text{tag}(\alpha) = \frac{\text{Tramo}(i, 11)}{\text{Tramo}(i, 4)}$$

$$\alpha = \arctg\left(\frac{\text{Tramo}(i, 11)}{\text{Tramo}(i, 4)}\right)$$

$$\text{tag}(\beta) = \frac{L_p - \text{Tramo}(i, 11)}{\text{Tramo}(i, 4)}$$

L_p = Longitud de paso.

$$L_p = \frac{\text{Tramo}(i, 3)}{\text{Tramo}(i, 2)}$$

$\text{Tramo}(i, 3)$ = Altura tramo.

$\text{Tramo}(i, 2)$ = N° de tramos.

Tenemos pues que el valor de β es igual a:

$$\beta = \arctg\left(\frac{\frac{Tramo(i,3)}{Tramo(i,2)} - Tramo(i,11)}{Tramo(i,4)}\right)$$

El resultado es que los montantes entre los nudos 2-10 y 4-12 se encuentran sometidos a una fuerza de $F_{Mont1} + F_{Mont2}$. El montante 2-10 estará a compresión y el montante 4-12 a tracción.

En los nudos centrales del paso en donde se unen las diagonales superiores e inferiores, cuando una diagonal está compresión se cumple que la otra está sometida a tracción. Como consecuencia tenemos que las componentes horizontales de F_{DiaSup} y F_{DiaInf} se anulen y las componentes verticales F_{Mont1} y F_{Mont2} se sumen. Esta última fuerza la absorbe el montante.

El resultado general es el siguiente:

- Montante 1-9 a compresión. F_{Mont1} y F_{Mont2}
- Montante 2-10 a compresión. F_{Mont1} y F_{Mont2}
- Montante 3-11 a tracción. F_{Mont1} y F_{Mont2}
- Montante 4-12 a tracción. F_{Mont1} y F_{Mont2}
- Diagonal superior a 2-7 a tracción. F_{DiaSup}
- Diagonal superior a 4-5 a compresión. F_{DiaInf}
- Diagonal inferior a 10-7 a compresión. F_{DiaSup}
- Diagonal inferior a 5-12 a tracción. F_{DiaInf}

Sabemos que en los montantes la sección y la fuerza es igual en todos ellos. La única diferencia es que en unos estarán a tracción y otros a compresión. Teniendo en cuenta que antes hemos despejado todas las incógnitas, tendremos pues:

$$F_{Mont} = F_{Mont1} + F_{Mont2}$$

$$F_{Mont} = F \cdot \tan \alpha + F \cdot \tan \beta$$

$$F_{Mont} = F \cdot \left(\frac{Tramo(i, 11)}{Tramo(i, 4)} + \frac{\frac{Tramo(i, 3)}{Tramo(i, 2)} - Tramo(i, 11)}{Tramo(i, 4)} \right)$$

$$F_{Mont} = F \cdot \left(\frac{Tramo(i, 3)}{Tramo(i, 4) \cdot Tramo(i, 2)} \right)$$

$$ES_{Mon} = \frac{F_{Mon}}{Tramo(i, 6)} = F \cdot \frac{Tramo(i, 3)}{Tramo(i, 4) \cdot Tramo(i, 2) \cdot Tramo(i, 6)}$$

$$ES_{Mon} = \frac{T \cdot Tramo(i, 3)}{2 \cdot Tramo(i, 4) \cdot Tramo(i, 2) \cdot Tramo(i, 6)}$$

$Tramo(i, 6)$ = Sección del Montante

Con las diagonales superiores ocurre lo mismo. Tendremos dos cargadas y otras dos descargadas. Una a tracción y otra a compresión.

$$F_{DiaSup} = \frac{F}{\cos(\alpha)}$$

$$ES_{DiaSup} = \frac{F}{\cos(\alpha) \cdot Tramo(i, 7)}$$

$$ES_{DiaSup} = \frac{T}{2 \cdot \cos(\alpha) \cdot Tramo(i, 7)}$$

$Tramo(i, 7)$ = Sección de la Diagonal Superior

La diagonal inferior es el mismo caso que el de la diagonal superior.

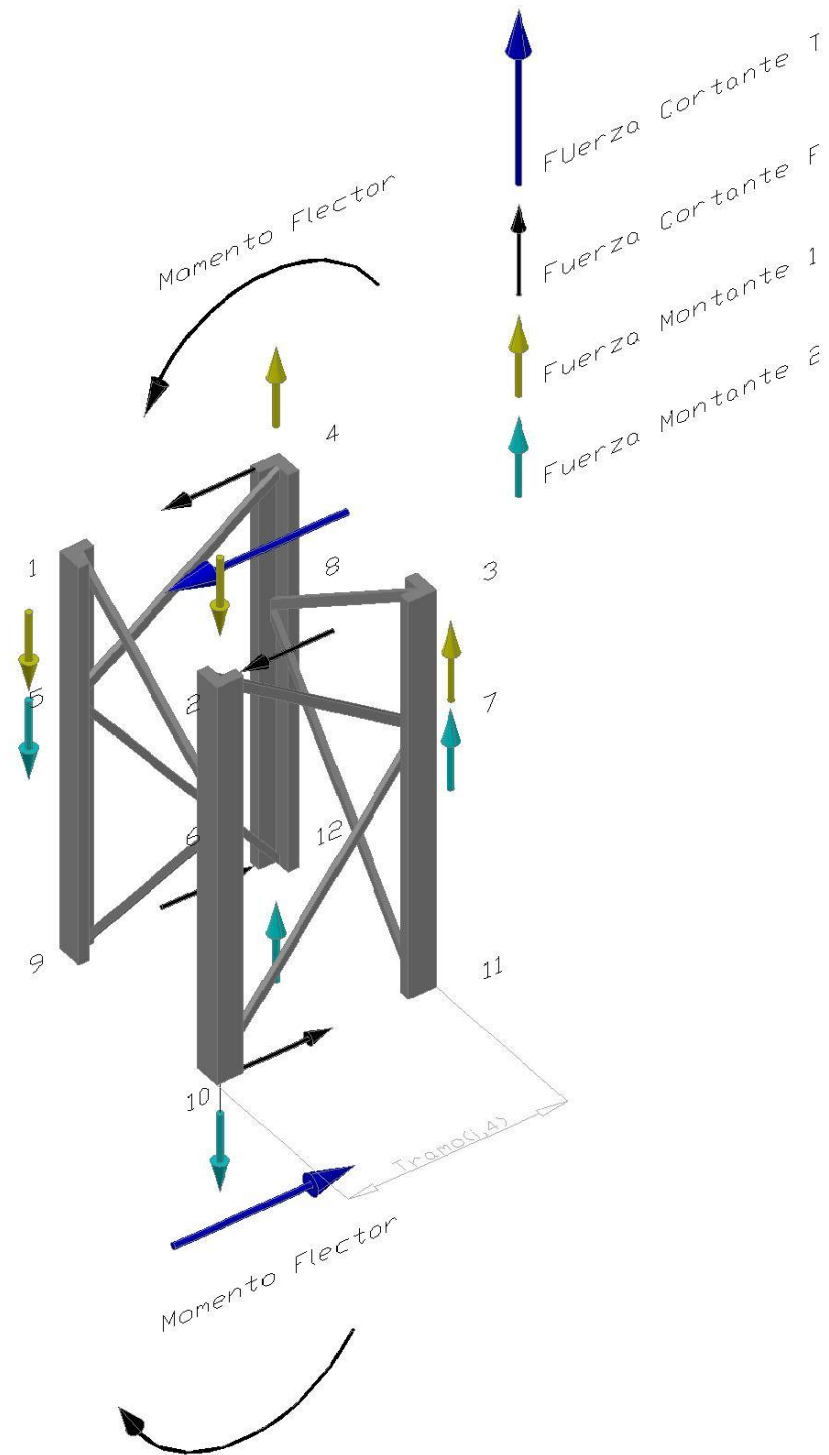
$$F_{DiaInf} = \frac{F}{\cos(\beta)}$$

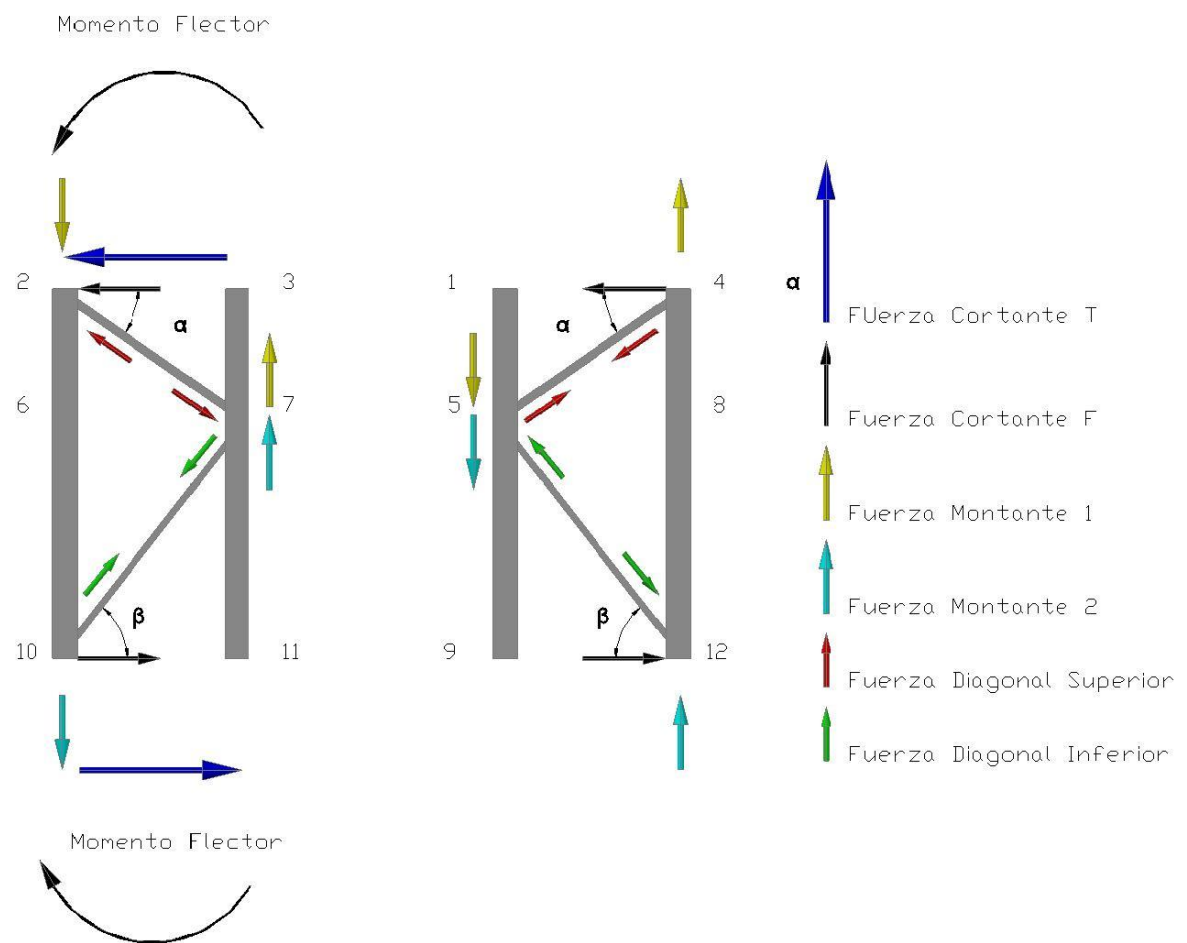
$$ES_{DiaInf} = \frac{F}{\cos(\beta) \cdot Tramo(i, 8)}$$

$$ES_{DiaInf} = \frac{T}{2 \cdot \cos(\beta) \cdot Tramo(i, 8)}$$

$Tramo(i, 8) =$ Sección de la Diagonal Inferior

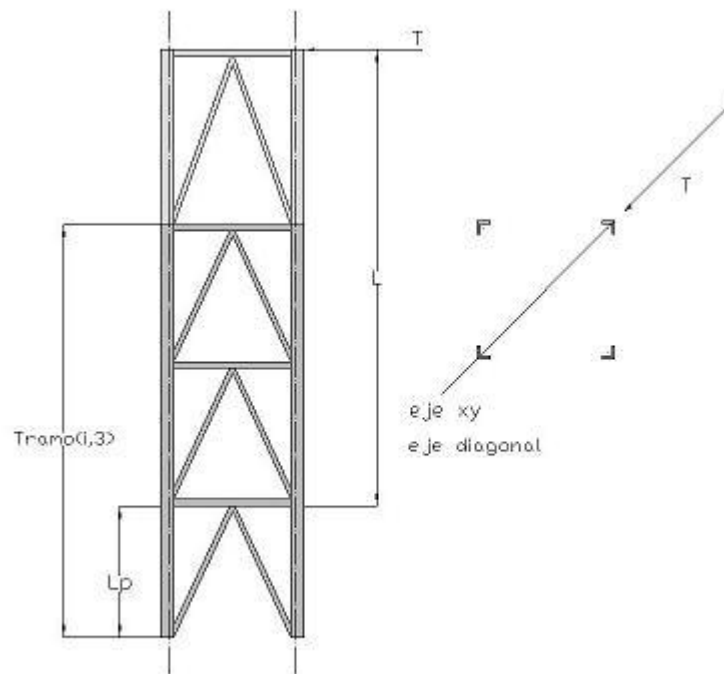
Debemos de comprobar que las fuerzas cortantes son las responsables de los momentos producidos por los montantes. Los esfuerzos en lo montantes son iguales que en el caso del tramo Tipo 1, y además que su geometría es igual en el caso de los montantes, así que la demostración es la misma en ambos.





9.4 MOMENTOS FLECTORES CON FUERZAS CORTANTES DE DIRECCIÓN DIAGONAL A LA SECCIÓN. CASO 2

Para calcular este caso, realizamos prácticamente los mismos pasos que hemos tomado con el Caso 1. Sólo que esta vez el viento incide sobre la torre en dirección diagonal, con lo que las reacciones en montantes y diagonales no serán las mismas. Supondremos como en el Caso 1 que tenemos una fuerza T y que hacemos el estudio de un paso de tramo cuya sección superior está a una distancia L y la distancia a la sección inferior es $L + L_p$. De manera que el momento flector en la parte superior e inferior son respectivamente:



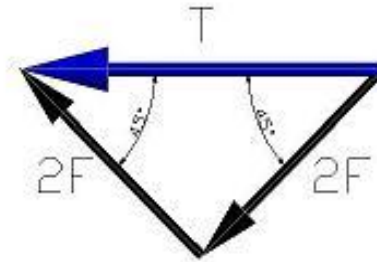
$$\text{Momento parte Superior} = T \cdot L$$

$$\text{Momento parte Inferior} = T \cdot (L + L_p)$$

9.4.1 Esfuerzos en tramo Tipo 1

Igual que en el Caso 1 descompondremos la fuerza cortante T , en fuerzas con la misma dirección que las caras del tramo de torre. La fuerza T al incidir sobre la torre en dirección diagonal, forma 45° tanto como con una cara del tramo como con la otra. Con lo que al descomponer esta fuerza según la siguiente figura, tendremos $2F$. Para que no haya

problemas de nomenclatura, daremos el doble de valor a F ya que cada una de estas fuerzas actuará en dos caras opuestas del tramo, así que en cada cara tenemos la mitad de esa fuerza, es decir F . Así conseguimos que podamos seguir utilizando las ecuaciones del anterior Caso 1, y no tener que estar dividiendo todos los valores por 2.

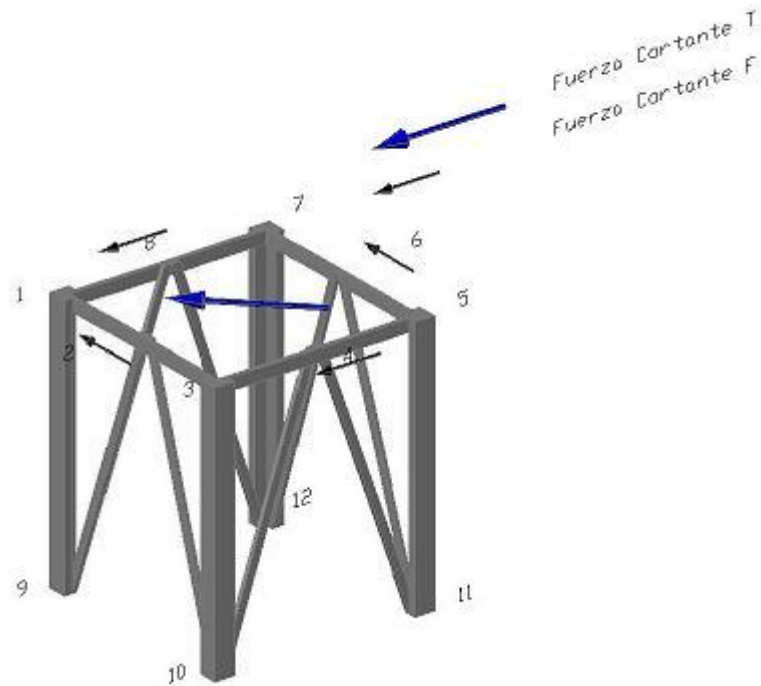


Si solucionamos este triángulo de fuerza tenemos que:

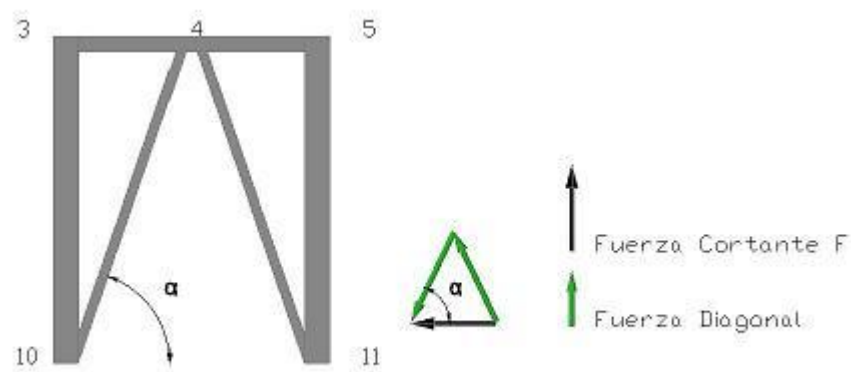
$$\sin(45) = \frac{2F}{T}; F = \frac{T \cdot \sin(45)}{2} = T \cdot \frac{1}{2} \cdot \frac{\sqrt{2}}{2} = \frac{T \cdot \sqrt{2}}{4} \cdot \frac{\sqrt{2}}{\sqrt{2}} = \frac{T \cdot 2}{4 \cdot \sqrt{2}} = \frac{T}{2 \cdot \sqrt{2}}$$

Con lo que la fuerza en cada cara de la torre será:

$$F = \frac{T}{2 \cdot \sqrt{2}}$$



Comprobamos que este caso es muy similar al Caso 1. De manera que si aplicamos las fuerzas en los nudos 2, 4, 6 y 8, vemos que obtenemos un triángulo de fuerzas idéntico al del Caso 1.



Simplemente siguiendo los pasos de la demostración del apartado anterior y aplicando la siguiente ecuación que hemos obtenido en el apartado anterior obtenemos las fuerzas en las diagonales inferiores. Unas estarán a tracción y otras a compresión, exactamente igual al caso anterior.

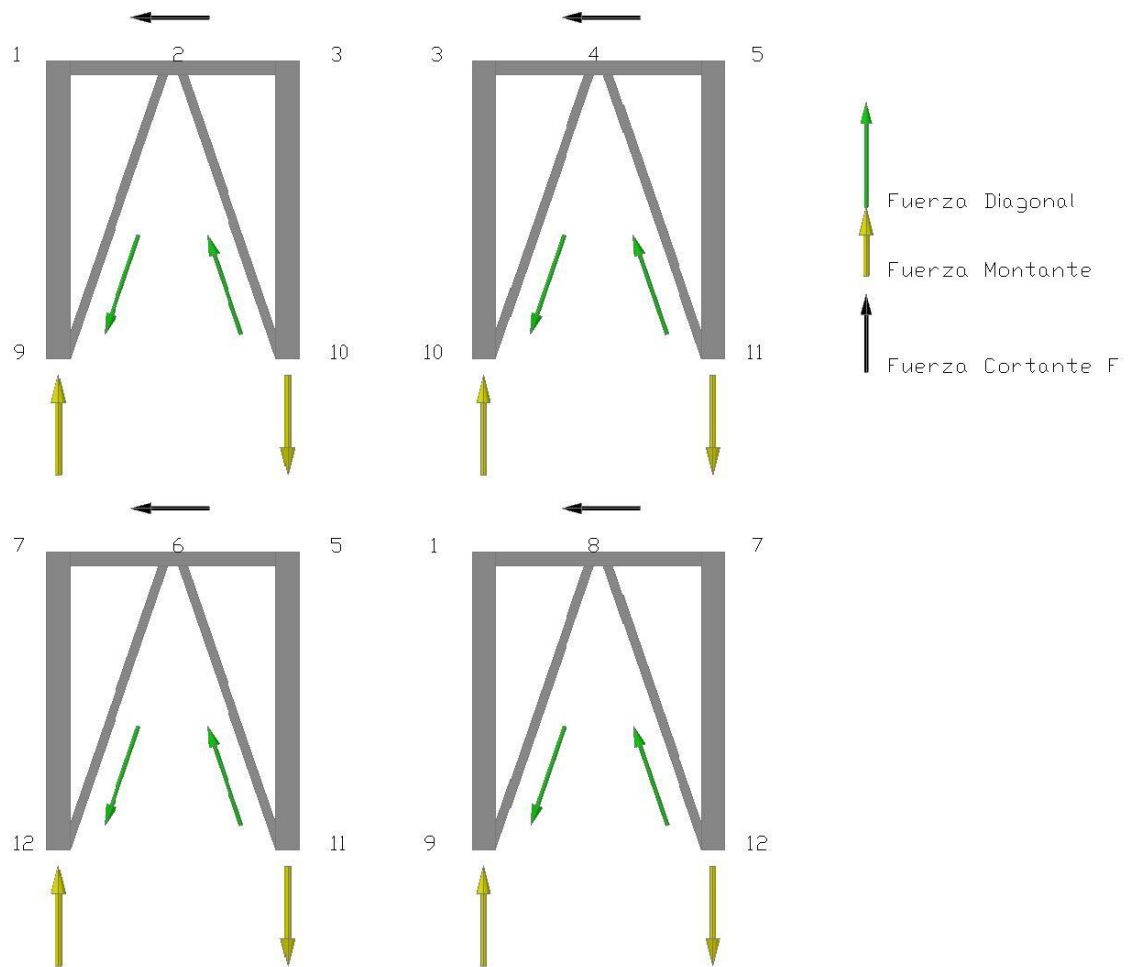
$$F_{Dialnf} = \frac{F}{2 \cdot \cos(\alpha)}$$

Obtenemos pues para todas las diagonales inferiores una fuerza, sustituyendo el valor de F

$$F_{Dialnf} = \frac{F}{2 \cdot \cos(\alpha)} = \frac{T}{2 \cdot \sqrt{2} \cdot 2 \cdot \cos(\alpha)} = \frac{T}{4 \cdot \sqrt{2} \cdot \cos(\alpha)}$$

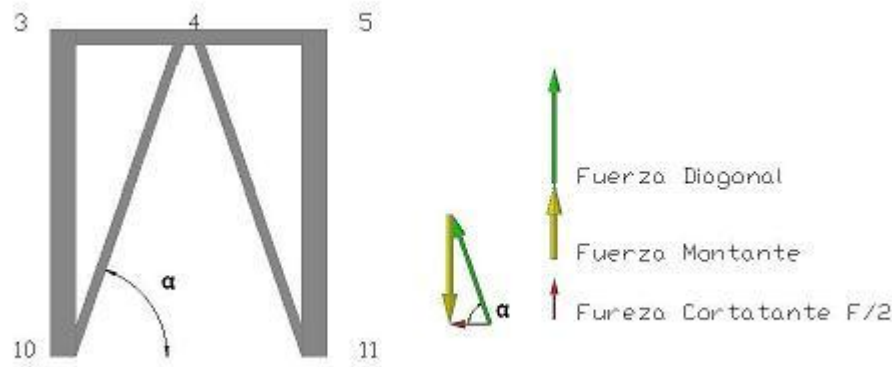
El cálculo de los montantes en este Caso 2 es algo distinto al anterior. Estos se cargan en los nudos 9, 10, 11 y 12 en que se unen dos diagonales inferiores con fuerza F_{Dialnf} . Pero en el anterior caso los montantes eran cargados únicamente por una única diagonal, así que simplemente aplicando equilibrio en el eje vertical obteníamos su valor. En este caso al estar cargados por dos diagonales inferiores tenemos el siguiente problema. Sabemos que todas las diagonales inferiores tienen el mismo modulo de fuerza, con lo que puede ser que en algún nudo las fuerzas de las diagonales inferiores se anulen entre ellas y que el montante este descargado.

Si estudiamos cada cara del tramo de torre por separado podemos comprobar el comportamiento de la estructura en global y comprobar cómo cada fuerza reacciona entre sí. Con la siguiente figura queda todo más claro.



Una vez que conocemos como actúan las diagonales podemos observar como en los nudos 10 y 12, las diagonales 2-10 y 4-10, 4-11 y 6-11 respectivamente, sus componentes verticales se anulan, con lo que los montantes 3-10 y 7-12 están descargados. En cambio los montantes 1-9 y 5-11 si que trabajan. Aplicando equilibrio en los nudos 9 y 11,

podemos ver como el estudio es igual al del Caso 1, solo que esta vez los montantes deben hacer frente al doble de la fuerza de las diagonales. Recordando la figura anterior que hemos explicado antes:



$$F_{Mon} = 2 \cdot F_{Dialnf} \cdot \text{Sen}(\alpha)$$

Si sustituimos el valor de F_{Dialnf} tendremos una fuerza en el montante de

$$F_{Mon} = 2 \cdot \frac{T}{4 \cdot \sqrt{2} \cdot \text{Cos}(\alpha)} \cdot \text{Sen}(\alpha) = \frac{T \cdot \tan(\alpha)}{2 \cdot \sqrt{2}}$$

Si recordamos el valor de $\tan(\alpha)$

$$F_{Mon} = \frac{T}{2 \cdot \sqrt{2}} \cdot \frac{2 \cdot \text{Tramo}(i, 3)}{\text{Tramo}(i, 2) \cdot \text{Tramo}(i, 4)} = \frac{T \cdot \text{Tramo}(i, 3)}{\sqrt{2} \cdot \text{Tramo}(i, 2) \cdot \text{Tramo}(i, 4)}$$

Haremos pues un pequeño resumen de cómo actúan las diagonales.

- Montante 1-9 a compresión.
- Montante 3-10 descargado.
- Montante 5-11 a tracción.
- Montante 7-12 descargado.
- Diagonal inferior 2-9 a compresión.
- Diagonal inferior 2-10 a tracción.
- Diagonal inferior 4-10 a compresión.
- Diagonal inferior 4-11 a tracción.

- Diagonal inferior 6-12 a compresión.
- Diagonal inferior 6-11 a tracción.
- Diagonal inferior 8-9 a compresión.
- Diagonal inferior 8-12 a compresión.

Una vez que tenemos todas las fuerzas de los elementos conseguimos sus esfuerzos simplemente dividiendo la fuerza por la sección.

$$ES_{Mon} = \frac{F_{Mon}}{Tramo(i, 6)} = \frac{T \cdot Tramo(i, 3)}{\sqrt{2} \cdot Tramo(i, 2) \cdot Tramo(i, 4) \cdot Tramo(i, 6)}$$

$Tramo(i, 6)$ = Sección del Montante

$$ES_{DiaInf} = \frac{F_{DiaInf}}{Tramo(i, 8)} = \frac{T}{4 \cdot \sqrt{2} \cdot \cos(\alpha) \cdot Tramo(i, 8)}$$

$Tramo(i, 8)$ = Sección de la Diagonal Inferior

Por último tenemos las diagonales superiores. Como he comentado antes es un único elemento de manera que absorbe todo el esfuerzo cortante, con lo que el esfuerzo en la diagonal superior, sustituyendo el valor de F será:

$$ES_{DiaSup} = \frac{F/2}{Tramo(i, 7)} = \frac{T}{4 \cdot \sqrt{2} \cdot Tramo(i, 7)}$$

$Tramo(i, 7)$ = Sección de la Diagonal Superior

Nuevamente realizamos un pequeño estudio para comprobar que son las fuerzas de los montantes los que contrarrestan los momentos producidos por las fuerzas cortantes. Calcularemos pues el momento producido por los montantes. En este caso hay dos montantes que pasa por el eje de giro, con lo que no generarán ningún momento. Estos montantes son 3-10 y 7-12. Con lo que los únicos montantes que contrarrestan los momentos provocados por las fuerzas cortantes son 1-9 y 5-11. Estos dos montantes tiene la misma fuerza, y la distancia al eje de giro es la misma para los dos con lo que tendremos que el momento resultante será:

$$M = 2 \cdot d \cdot F_{Mon}$$

$$d = \frac{Tramo(i, 4)}{\sqrt{2}}$$

$Tramo(i, 4) = \text{Lado del Tramo}$

$$M = 2 \cdot \frac{Tramo(i, 4)}{\sqrt{2}} \cdot \frac{T \cdot Tramo(i, 3)}{\sqrt{2} \cdot Tramo(i, 2) \cdot Tramo(i, 4)} = \frac{T \cdot Tramo(i, 3)}{Tramo(i, 2)}$$

Teniendo en cuenta que:

$$\frac{Tramo(i, 3)}{Tramo(i, 2)} = L_p$$

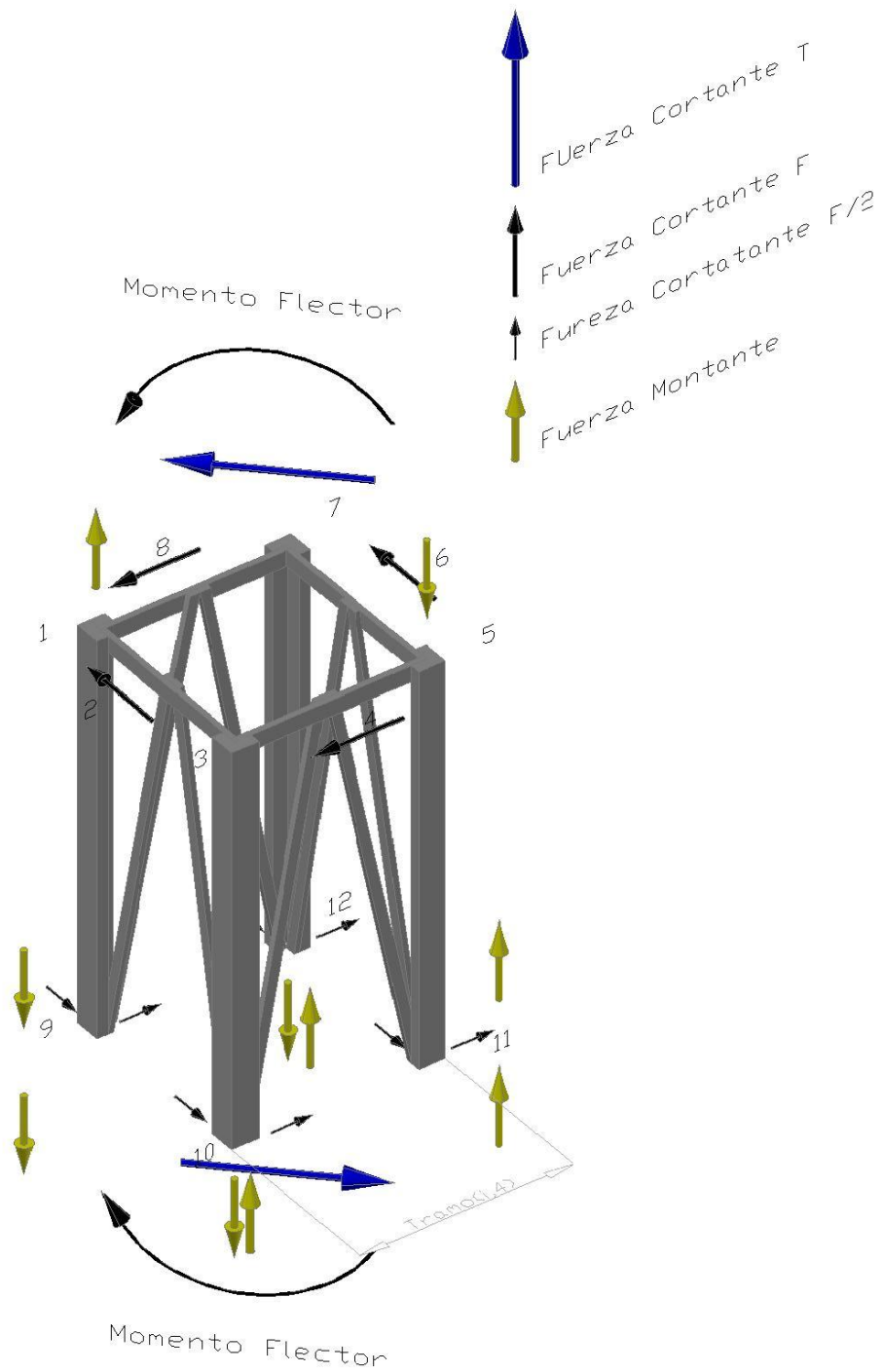
Tendremos que:

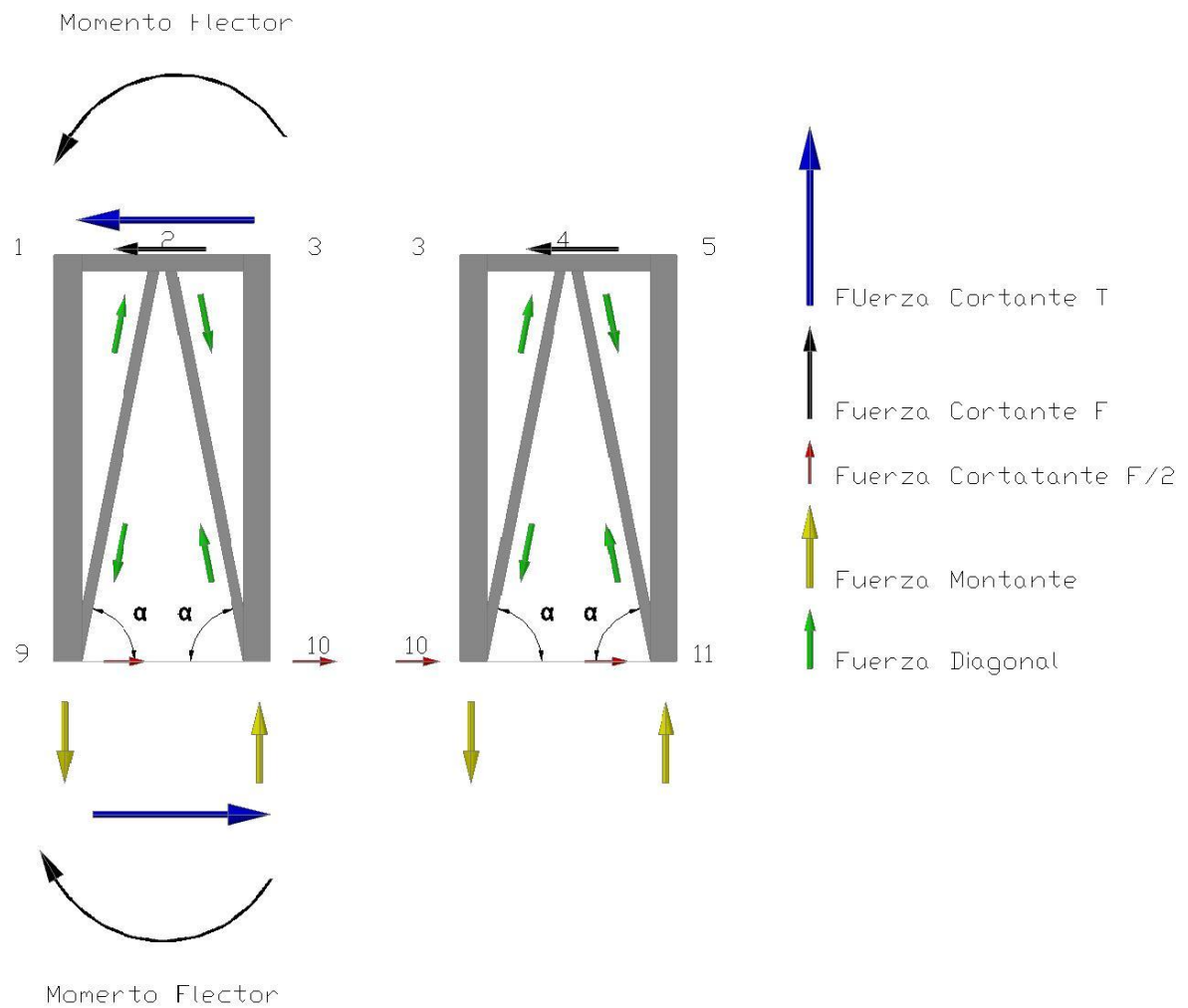
$$M = T \cdot L_p$$

Así pues que las fuerzas de los montantes producirán un momento igual al esfuerzo cortante por la diferencia de alturas entre la sección superior e inferior de cada paso de tramo. Como hemos partido de la teoría que los esfuerzos cortantes son absorbidos por las diagonales, las fueras de los montantes serán las encargadas de contrarrestar los momentos, otra de las teorías que hemos enunciado. Los tramos que se encuentran por encima del estudiado transmiten a los montantes unas fuerzas que producen el momento en la parte superior e inferior respectivamente

$$\text{Momento parte Superior} = T \cdot L$$

$$\text{Momento parte Inferior} = T \cdot (L + L_p)$$

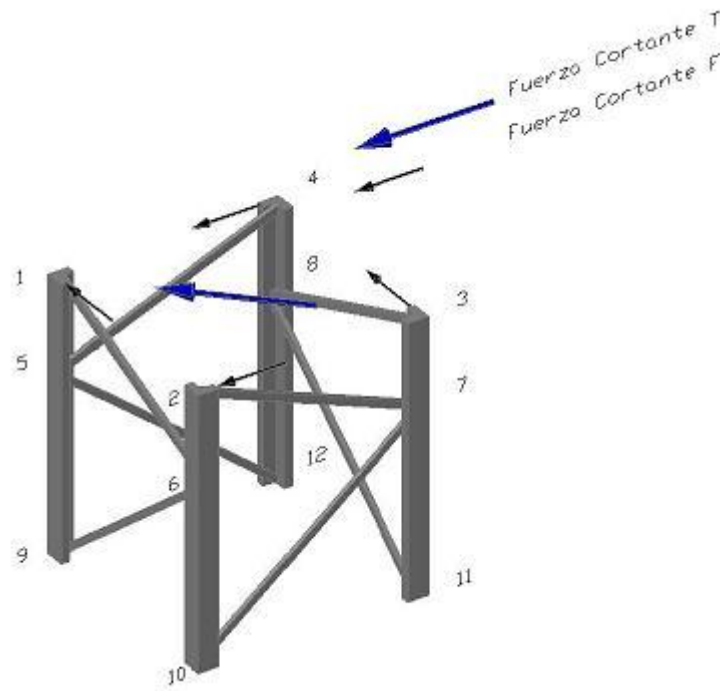




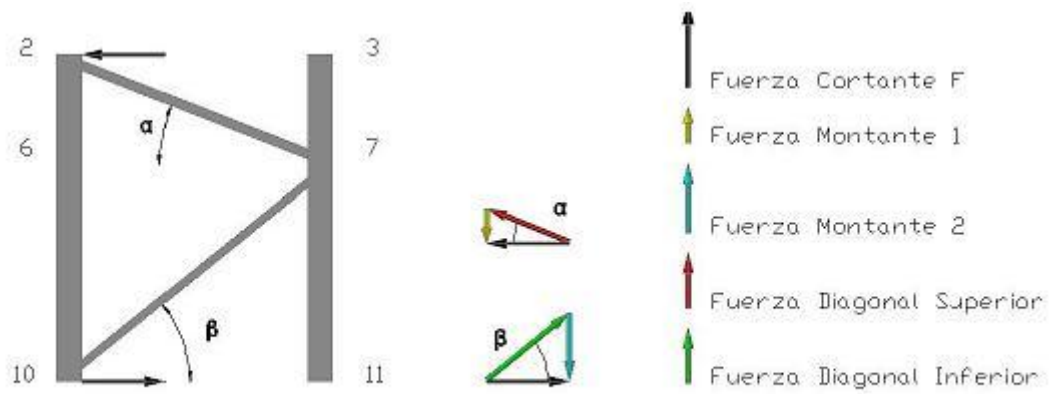
9.4.2 Esfuerzos en tramo Tipo 2

La descomposición de fuerzas cortantes es igual que en los tramos de Tipo 1, con lo que tenemos una fuerza F actuando en cada cara del paso. El valor de esta fuerza F lo hemos calculado anteriormente.

$$F = \frac{T}{2 \cdot \sqrt{2}}$$



Debemos de descomponer pues estas fuerzas F . Están aplicadas en los nudos 1,2,3 y 4. Con esta descomposición comprobamos que es idéntica al caso 9.3.2 Caso 1, ya que tenemos una geometría de tramos iguales. La única diferencia es que en este caso todas las diagonales estarán cargadas. Con los valores que hemos hallado en el Caso 1 tenemos todos los valores para poder conseguir los esfuerzos en las diagonales superiores e inferiores.



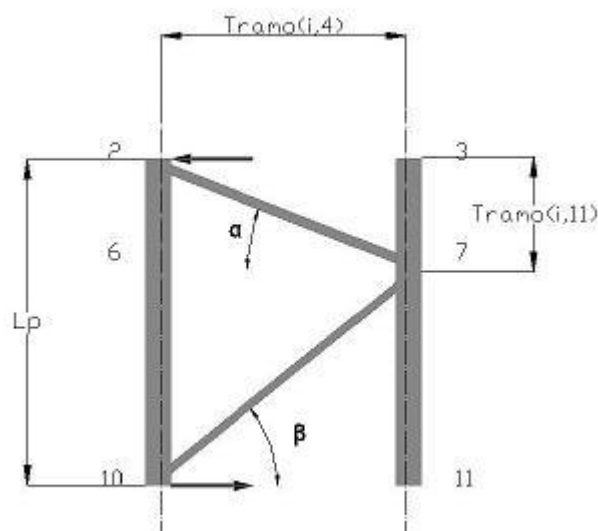
$$F_{Mont1} = F \cdot \tan \alpha$$

$$F_{DiaSup} = \frac{F}{\cos(\alpha)}$$

$$F_{Mont2} = F \cdot \tan \beta$$

$$F_{DiaInf} = \frac{F}{\cos(\beta)}$$

Los ángulos se calculan de la misma manera.



$$\tan(\alpha) = \frac{\text{Tramo}(i, 11)}{\text{Tramo}(i, 4)}$$

$$\alpha = \arctg\left(\frac{\text{Tramo}(i, 11)}{\text{Tramo}(i, 4)}\right)$$

$$\tan(\beta) = \frac{L_p - \text{Tramo}(i, 11)}{\text{Tramo}(i, 4)}$$

L_p = Longitud de paso.

$$L_p = \frac{\text{Tramo}(i, 3)}{\text{Tramo}(i, 2)}$$

$\text{Tramo}(i, 3)$ = Altura tramo.

$\text{Tramo}(i, 2)$ = N° de tramos.

Tenemos pues que el valor de β es igual a:

$$\beta = \arctg\left(\frac{\frac{\text{Tramo}(i, 3)}{\text{Tramo}(i, 2)} - \text{Tramo}(i, 11)}{\text{Tramo}(i, 4)}\right)$$

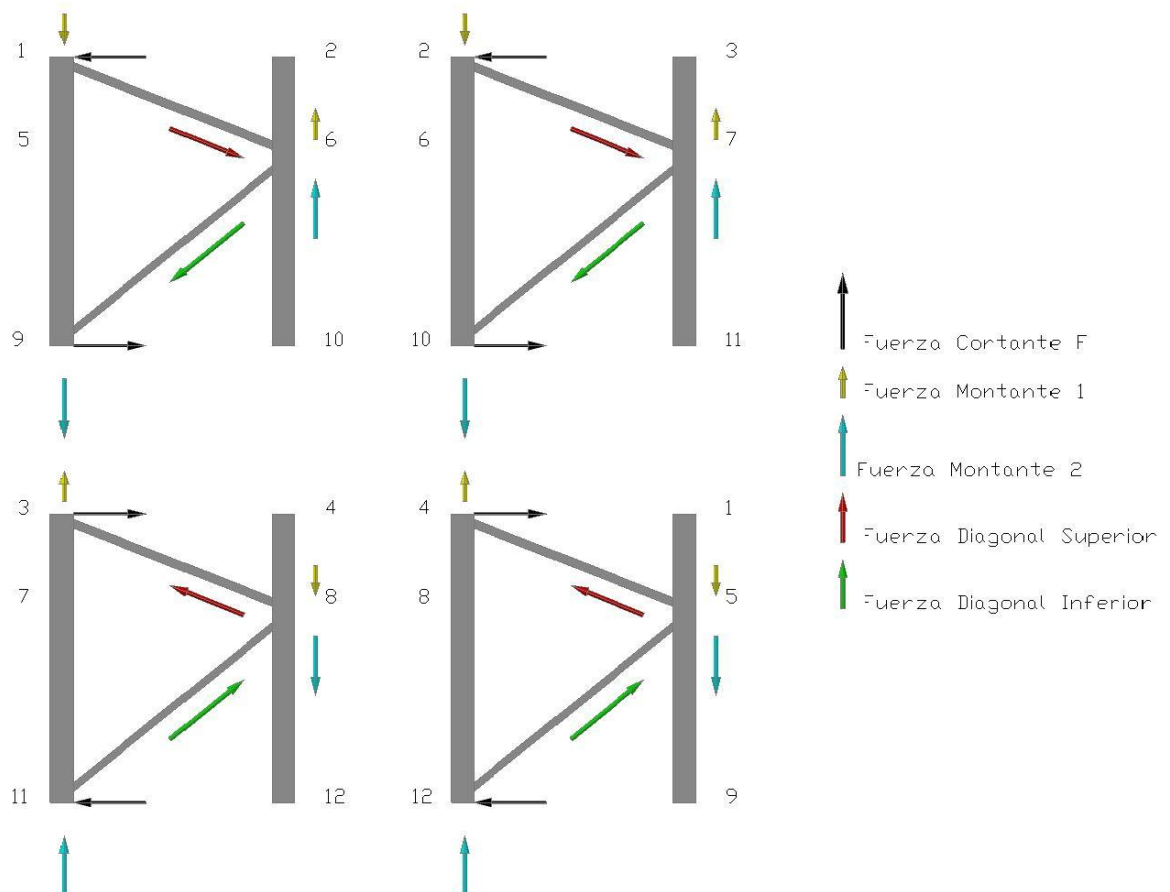
Con estas ecuaciones hallamos todas las fuerzas que resultan del sumatorio de fuerzas en los nudos 1,2,3,4,9,10,11 y 12. Estos nudos son los superiores e inferiores del paso del tramo.

Los nudos 5,6,7 y 8 son los únicos que aún no se han estudiado. En estos nudos se unen diagonal superior, diagonal inferior y montante. En estos nudos conocemos las fuerzas de las diagonales superiores e inferiores, pero desconocemos la del montante. Tenemos como condición que cuando la diagonal superior trabaja a tracción la diagonal inferior trabaja a compresión y viceversa, con lo que como las componentes horizontales de estas fuerzas horizontales son iguales se anulan, pero las componentes verticales se suman. De este sumatorio de fuerza sacamos la fuerza en el montante.

Como conocemos el valor de estas fuerzas verticales que son F_{Mont1} y F_{Mont2} , con lo que en los nudos centrales se aplica sobre el montante una fuerza F_{Mont} igual a:

$$F_{Mont} = F_{Mont1} + F_{Mont2}$$

Una vez que sabemos cómo actúan las fuerzas en los nudos, sabemos como va a quedar cargada cada barra. Con las siguientes figuras todo queda más claro. En el que estudiamos cada cara por separado.



El problema lo tenemos en los montantes, ya que una parte queda a compresión y otra a tracción, con lo que estudiaremos cada montante por separado.

- Montante 1-9

Este montante está totalmente a compresión. En la base tenemos una fuerza F_{Mont2} pero va creciendo hasta un valor de $F_{Mont1} + F_{Mont2}$ en el nudo superior.

- Montante 2-10

Este montante se encuentra dividido. Tenemos una parte a tracción y otra a compresión. La parte superior, la de los nudos 2-6 se encuentra con una fuerza F_{Mont1} a tracción. La parte inferior, nudos 6-10 a compresión con una fuerza F_{Mont2} .

- Montante 3-11

Este montante está totalmente a tracción. En la parte superior tenemos una fuerza F_{Mont1} pero va creciendo hasta un valor de $F_{Mont1} + F_{Mont2}$ en el nudo inferior.

- Montante 4-12

Este montante se encuentra dividido. Tenemos una parte a tracción y otra a compresión. La parte superior, la de los nudos 4-8 se encuentra con una fuerza F_{Mont1} a tracción. La parte inferior, nudos 8-12 a compresión con una fuerza F_{Mont2} .

Calcularemos los esfuerzos.

Las diagonales superiores son todas iguales, unas a tracción y otras a compresión.

$$F_{DiaSup} = \frac{F}{\cos(\alpha)}$$

$$Es_{DiaSup} = \frac{F}{\cos(\alpha) \cdot Tramo(i, 7)}$$

$$Es_{DiaSup} = \frac{T}{2 \cdot \sqrt{2} \cdot \cos(\alpha) \cdot Tramo(i, 7)}$$

$Tramo(i, 7)$ = Sección de la Diagonal Superior

La diagonal inferior es el mismo caso que el de la diagonal superior.

$$F_{DialInf} = \frac{F}{\cos(\beta)}$$

$$ES_{DialInf} = \frac{F}{\cos(\beta) \cdot Tramo(i, 8)}$$

$$ES_{DialInf} = \frac{T}{2 \cdot \sqrt{2} \cdot \cos(\beta) \cdot Tramo(i, 8)}$$

$Tramo(i, 8)$ = Sección de la Diagonal Inferior

En el caso del estudio de los montantes hemos visto que tenemos distintos casos de estudio. Pero como hasta ahora estudiaremos el caso más desfavorable, que es el que nos interesa. Tenemos dos casos el del montante 1-9 que esta a tracción y el del montante 3-11 que esta a compresión. Los dos tendrán el mismo valor.

$$F_{Mont} = 2 \cdot (F_{Mont1} + F_{Mont2})$$

Sustituimos valores.

$$F_{Mont} = 2 \cdot F \cdot (\tan \alpha + \tan \beta)$$

$$F_{Mont} = 2 \cdot F \cdot \left(\frac{Tramo(i, 11)}{Tramo(i, 4)} + \frac{\frac{Tramo(i, 3)}{Tramo(i, 2)} - Tramo(i, 11)}{Tramo(i, 4)} \right)$$

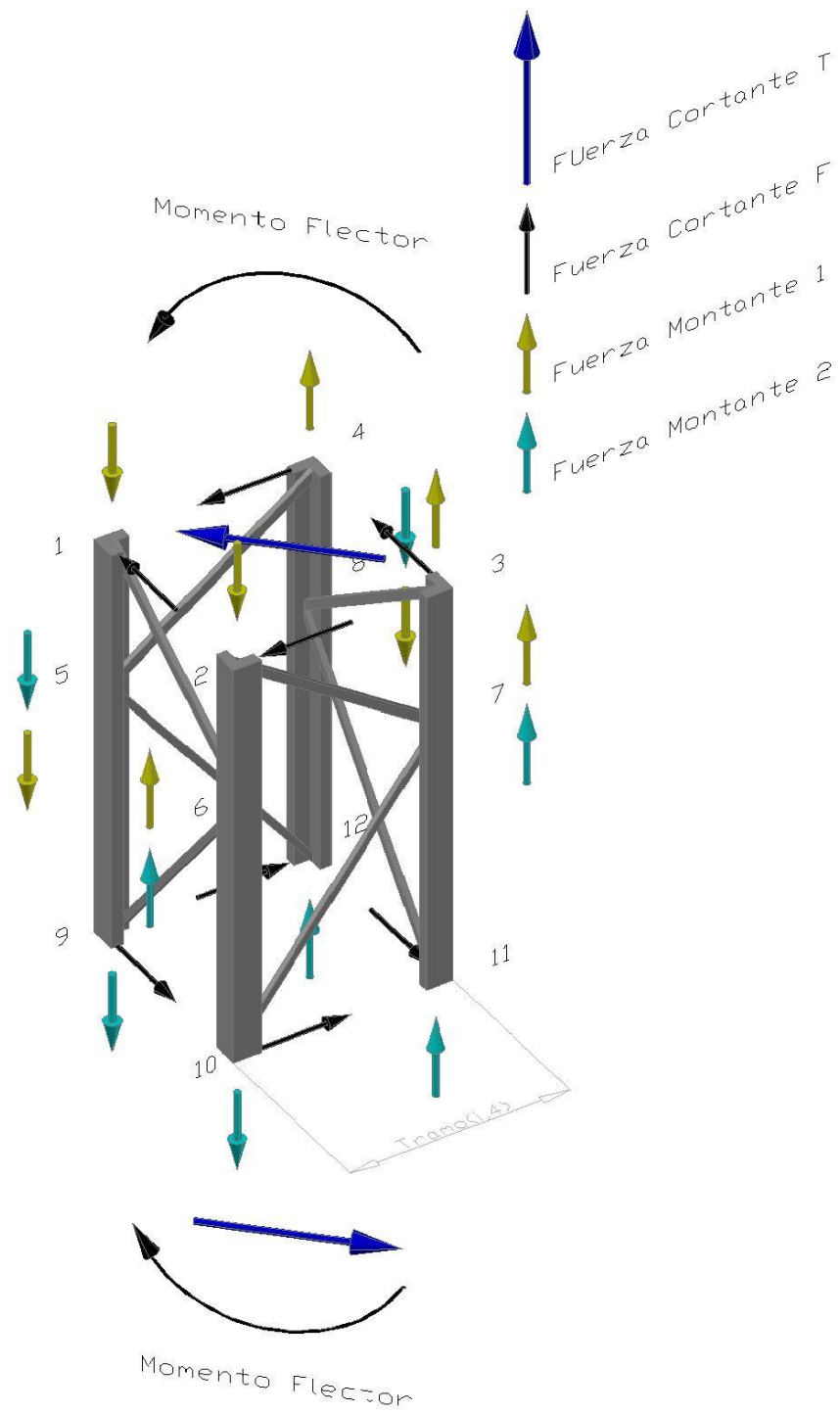
$$F_{Mont} = 2 \cdot F \cdot \left(\frac{Tramo(i, 3)}{Tramo(i, 4) \cdot Tramo(i, 2)} \right)$$

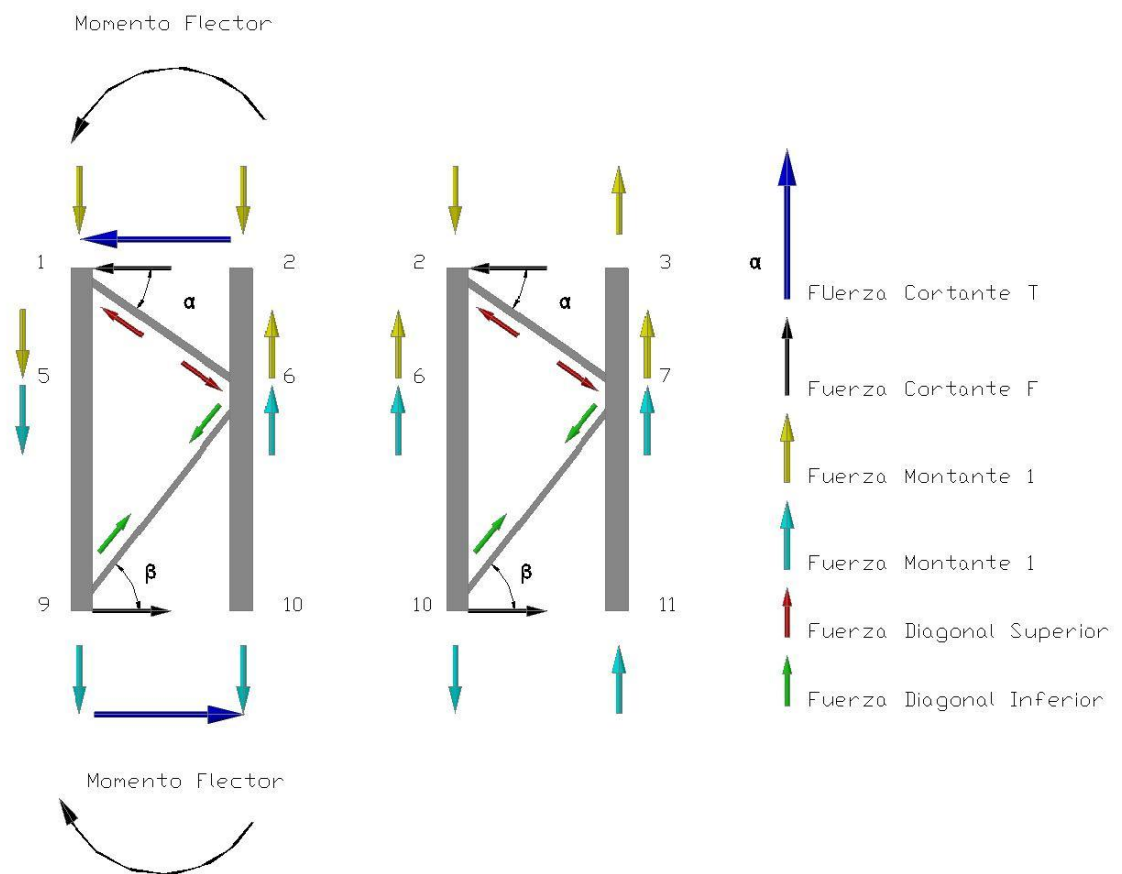
$$ES_{Mon} = \frac{F_{Mon}}{Tramo(i, 6)} = 2 \cdot \frac{T}{2 \cdot \sqrt{2}} \cdot \frac{Tramo(i, 3)}{Tramo(i, 4) \cdot Tramo(i, 2) \cdot Tramo(i, 6)}$$

$$Es_{Mon} = \frac{T \cdot Tramo(i, 3)}{\sqrt{2} \cdot Tramo(i, 4) \cdot Tramo(i, 2) \cdot Tramo(i, 6)}$$

$Tramo(i, 6)$ = Sección del Montante

La comprobación del momento que producen las fuerzas en los montantes es igual que en los tramos del Tipo 1.





9.5 RESUMEN ESFUERZOS.

Hemos calculado todos los esfuerzos de los montantes y diagonales según el caso de estudio. Haremos un pequeño resumen para que quede todo claro.

1.- Flexión pura en dirección transversal a la sección. Caso 1

$$\text{Montante} - Es_{Mon} = \frac{M_{Flec}}{2 \cdot \text{Tramo}(i, 4) \cdot \text{Tramo}(i, 6)}$$

Diagonal Superior Descargada

Diagonal Inferior Descargada

2.- Flexión pura en dirección diagonal a la sección. Caso 2

$$\text{Montante} - Es_{Mon} = \frac{M_{Flec}}{\sqrt{2} \cdot \text{Tramo}(i, 4) \cdot \text{Tramo}(i, 6)}$$

Diagonal Superior Descargada

Diagonal Inferior Descargada

3.- Momentos flectores con fuerzas cortantes en dirección transversal a la sección. Caso 1

Tramo Tipo 1

$$\text{Montant} - Es_{Mon} = \frac{T \cdot \text{Tramo}(i, 3)}{2 \cdot \text{Tramo}(i, 2) \cdot \text{Tramo}(i, 4) \cdot \text{Tramo}(i, 6)}$$

$$\text{Diagonal Superior} - Es_{DiaSup} = \frac{T}{4 \cdot \text{Tramo}(i, 7)}$$

$$\text{Diagonal Inferior} - Es_{DiaInf} = \frac{T}{4 \cdot \cos(\alpha) \cdot \text{Tramo}(i, 8)}$$

Tramo Tipo 2

$$\text{Montant} - Es_{Mon} = \frac{T \cdot \text{Tramo}(i, 3)}{2 \cdot \text{Tramo}(i, 2) \cdot \text{Tramo}(i, 4) \cdot \text{Tramo}(i, 6)}$$

$$\text{Diagonal Superior} - Es_{DiaSup} = \frac{T}{2 \cdot \cos(\alpha) \cdot \text{Tramo}(i, 7)}$$

$$\text{Diagonal Inferior} - Es_{DiaInf} = \frac{T}{2 \cdot \cos(\beta) \cdot \text{Tramo}(i, 8)}$$

4.- Momentos flectores con fuerzas cortantes de dirección transversal a la sección.
 Caso 2

Tramo Tipo 1

$$\text{Montant} - Es_{Mon} = \frac{T \cdot \text{Tramo}(i, 3)}{\sqrt{2} \cdot \text{Tramo}(i, 2) \cdot \text{Tramo}(i, 4) \cdot \text{Tramo}(i, 6)}$$

$$\text{Diagonal Superior} - Es_{DiaSup} = \frac{T}{4 \cdot \sqrt{2} \cdot \text{Tramo}(i, 7)}$$

$$\text{Diagonal Inferior} - Es_{DiaInf} = \frac{T}{4 \cdot \sqrt{2} \cdot \cos(\alpha) \cdot \text{Tramo}(i, 8)}$$

Tramo Tipo 2

$$\text{Montant} - Es_{Mon} = \frac{T \cdot \text{Tramo}(i, 3)}{\sqrt{2} \cdot \text{Tramo}(i, 2) \cdot \text{Tramo}(i, 4) \cdot \text{Tramo}(i, 6)}$$

$$\text{Diagonal Superior} - Es_{DiaSup} = \frac{T}{2 \cdot \sqrt{2} \cdot \cos(\alpha) \cdot \text{Tramo}(i, 7)}$$

$$\text{Diagonal Inferior} - Es_{DiaInf} = \frac{T}{2 \cdot \sqrt{2} \cdot \cos(\beta) \cdot \text{Tramo}(i, 8)}$$

Una vez que tenemos claro que esfuerzos encontramos en cada montante y diagonal según el caso de estudio, seguiremos con el proyecto.

Nuestros cálculos para que sean precisos están basados en que las fuerzas están aplicadas en cada nudo de la torre. Pero al dividir la torre en incrementos sólo sabemos que estos esfuerzos están aplicados exactamente en unos nudos en concreto, es decir en los nudos de la parte inferior del tramo. Los resultados son reflejados para cada tramo, pero como en cada tramo tenemos varios pasos de tramo, tomamos como valor representativo el paso más bajo de todos ellos, y con ello utilizaremos el valor del incremento más bajo. Tendremos pues con esto los resultados más altos de todo el tramo de torre. Es decir los más desfavorables y representativos.

Queda claro que todos los esfuerzos que hemos encontrado son directamente proporcionales tanto a los momentos flectores, en el cálculo de los montantes, como a los esfuerzos cortantes, en el caso de las diagonales superiores e inferiores. Por ello creamos una nueva variable por cada esfuerzo que hemos encontrado. Esta variable es el de los esfuerzos producidos por una fuerza o momento unitario.

Simplemente aplicando estos esfuerzos unitarios en la base de cada tramo de torre obtendremos nuestros resultados.

Creamos pues nuevas variables $EsfUnit_{Mon}(3)$, $EsfUnit_{DS}(3)$ y $EsfUnit_{DI}(3)$

$EsfUnit_{Mont}(1)$ Esfuerzos en el montante por flexión pura en dirección transversal a la sección. Caso 1

$$ES_{Mon} = \frac{M_{Flec}}{2 \cdot Tramo(i, 4) \cdot Tramo(i, 6)}$$

$$EsfUnit_{Mont}(1) = \frac{1}{2 \cdot Tramo(i, 4) \cdot Tramo(i, 6)}$$

$EsfUnit_{DS}(1)$ Esfuerzos en la diagonal superior por esfuerzos cortante en dirección transversal a la sección. Caso 1

Tramo Tipo 1

$$ES_{DiaSup} = \frac{T}{4 \cdot Tramo(i, 7)}$$

$$EsfUnit_{DS}(1) = \frac{1}{4 \cdot Tramo(i, 7)}$$

Tramo Tipo 2

$$ES_{DiaSup} = \frac{T}{2 \cdot \cos(\alpha) \cdot Tramo(i, 7)}$$

$$EsfUnit_{DS}(1) = \frac{1}{2 \cdot \cos(\alpha) \cdot Tramo(i, 7)}$$

$EsfUnit_{DI}(1)$ Esfuerzos en la diagonal inferior por esfuerzos cortante en dirección transversal a la sección. Caso 1

Tramo Tipo 1

$$ES_{Dialnf} = \frac{T}{4 \cdot \cos(\alpha) \cdot Tramo(i, 8)}$$

$$EsfUnit_{DI}(1) = \frac{1}{4 \cdot \cos(\alpha) \cdot Tramo(i, 8)}$$

Tramo Tipo 2

$$ES_{Dialnf} = \frac{T}{2 \cdot \cos(\beta) \cdot Tramo(i, 8)}$$

$$EsfUnit_{DI}(1) = \frac{1}{2 \cdot \cos(\beta) \cdot Tramo(i, 8)}$$

$EsfUnit_{Mont}(2)$ Esfuerzos en el montante por flexión pura en dirección diagonal a la sección. Caso 2

$$ES_{Mon} = \frac{M_{Flec}}{\sqrt{2} \cdot Tramo(i, 4) \cdot Tramo(i, 6)}$$

$$EsfUnit_{Mont}(2) = \frac{1}{\sqrt{2} \cdot Tramo(i, 4) \cdot Tramo(i, 6)}$$

$EsfUnit_{DS}(2)$ Esfuerzos en la diagonal superior por esfuerzos cortante en dirección diagonal a la sección. Caso 2

Tramo Tipo 1

$$ES_{DiaSup} = \frac{T}{4 \cdot \sqrt{2} \cdot Tramo(i, 7)}$$

$$EsfUnit_{DS}(2) = \frac{1}{4 \cdot \sqrt{2} \cdot Tramo(i, 7)}$$

Tramo Tipo 2

$$ES_{DiaSup} = \frac{T}{2 \cdot \sqrt{2} \cdot \cos(\alpha) \cdot Tramo(i, 7)}$$

$$EsfUnit_{DS}(2) = \frac{1}{2 \cdot \sqrt{2} \cdot \cos(\alpha) \cdot Tramo(i, 7)}$$

$EsfUnit_{DI}(2)$ Esfuerzos en la diagonal inferior por esfuerzos cortante en dirección diagonal a la sección. Caso 2

Tramo Tipo 1

$$ES_{Dialnf} = \frac{T}{4 \cdot \sqrt{2} \cdot \cos(\alpha) \cdot Tramo(i, 8)}$$

$$EsfUnit_{DI}(2) = \frac{1}{4 \cdot \sqrt{2} \cdot \cos(\alpha) \cdot Tramo(i, 8)}$$

Tramo Tipo 2

$$ES_{Dialnf} = \frac{T}{2 \cdot \sqrt{2} \cdot \cos(\beta) \cdot Tramo(i, 8)}$$

$$EsfUnit_{DI}(2) = \frac{1}{2 \cdot \sqrt{2} \cdot \cos(\beta) \cdot Tramo(i, 8)}$$

Hasta ahora no hemos tenido en cuenta el peso de la estructura en sí. Supondremos que el peso de la estructura se reparte entre los 4 montantes, sin tener en cuenta el peso de las diagonales. Debemos de tener en cuenta también el peso de la góndola y el peso de las aspas. De manera que si repartimos el peso entre los cuatro montantes el esfuerzo en cada montante dependiendo del incremento en el que estemos haciendo el cálculo será:

$$EsfUnit_{Mont}(3) = \frac{(Peso(j) + Dat(2) + Dat(3)) \cdot Dat(10)}{4 \cdot Tramo(i, 6)}$$

Como el esfuerzo es proporcional al peso del incremento y peso de la góndola y aspas tenemos que:

$$EsfUnit_{Mont}(3) = \frac{1}{4 \cdot Tramo(i, 6)}$$

$Tramo(i, 6)$ = Sección del Montante

Una vez que hemos explicado como obtenemos los esfuerzos unitarios de montantes y diagonales para los dos casos de estudio, determinaremos que es lo que el programa de cálculo hace.

Hemos comentado anteriormente que los resultados que obtendremos son los referidos a la base de cada tramo de torre. Estos siempre son los más desfavorables. El programa realiza un bucle desde el primer hasta el último tramo de torre, para obtener estos esfuerzos unitarios. Con estos esfuerzos logramos los siguientes resultados, que guardamos en las variables aux(2) y T(9,2). A continuación se detalla cómo se obtienen los resultados para el Caso 1, siendo para el Caso 2 exactamente igual el proceso de cálculo, pero con sus respectivos datos.

En la variable aux(2), guardamos los momentos secundarios que hay en el incremento que se está estudiando. Cada espacio de memoria corresponde a un caso de estudio.

En la variable T(9,2) guardamos 9 distintos resultados. El segundo vector lo utilizamos para guardar los resultados de los dos casos de estudio por separado.

- T(1,1): Número de tramo
- T(2,1): Altura respecto del suelo
- T(3,1): Peso
- T(4,1): Fuerza cortante
- T(5,1): Momento total
- T(6,1): Flecha
- T(7,1): Tensión en el montante N/m^2
- T(8,1): Tensión diagonal superior N/m^2
- T(9,1): Tensión diagonal inferior N/m^2

Se ha comentado que el programa mediante un bucle desde el primer hasta el último tramo de torre realiza los cálculos. Pero no solo con esto indicamos que incremento del tramo es el que se quiere calcular. Para ello utilizamos la variable ki . A esta variable se le da siempre el valor del último incremento de cada tramo. La obtenemos de la siguiente manera

$$ki = i \cdot NInc(1)$$

Siendo:

i Número de tramo que se está estudiando

$NInc(1)$ Número de incrementos que tiene cada tramo

Las tensiones en los montantes y diagonales las obtenemos simplemente multiplicando el momento o esfuerzo cortante por los esfuerzos unitarios que hemos explicando anteriormente. Los resultados lo obtenemos de la siguiente manera.

Los esfuerzos de los montantes y diagonales se dividen entre 10000 para tener las unidades en metros.

$$\begin{aligned} aux(1) = & MomentosSecunCaso1(ki, 1) + MomentosSecunCaso1(ki, 2) \\ & - Abs(MomentosSecunCaso1(ki, 3)) + MomentosSecunCaso1(ki, 4) \end{aligned}$$

$$T(1,1) = i$$

$$T(2,1) = Altura$$

$$T(3,1) = (Dat(2) + Dat(3) + Peso(ki)) * Dat(10)$$

$$T(4,1) = FVientoElementos(1) + FVientoElementos(2) + EsfCortanteVientoCaso1(ki)$$

$$\begin{aligned} T(5,1) = & MomentosCaso1(ki, 1) + MomentosCaso1(ki, 2) - MomentosCaso1(ki, 3) \\ & + MomentosCaso1(ki, 4) + aux(1) \end{aligned}$$

$$\begin{aligned} T(6,1) = & FlechasCaso1(ki, 1) + FlechasCaso1(ki, 2) + FlechasCaso1(ki, 3) \\ & + FlechasCaso1(ki, 4) \end{aligned}$$

$$T(7,1) = T(5, 1) * EsfUnitMont(1)/10000 + T(3, 1) * EsfUnitMont(3)/10000$$

$$T(8,1) = T(4, 1) * EsfUnitDS(1)/10000$$

$$T(9,1) = T(4, 1) * EsfUnitDI(1)/10000$$

Una vez obtenido los resultados pasamos estos valores a la tabla del formulario.

9.6 EXPLICACIÓN RUTINA.

Declaración de variables

Dim i As Integer

Dim ii As Integer

Dim j As Integer

Dim ki As Integer

Dim l As Integer

Dim EsfUnitMont(6) As Double

Dim EsfUnitDS(4) As Double

```
Dim EsfUnitDI(4) As Double
Dim MOM2Caso1(4) As Double
Dim MOM2Caso2(4) As Double
Dim T(9, 4) As Double
Dim aux(3) As Single
```

En el formulario de cálculo de torre a las tablas les daremos un valor filas igual al número tramos de torre más 2.

```
frmcalculotorre.Grid1(1).Rows = Dat(1) + 2
frmcalculotorre.Grid1(2).Rows = Dat(1) + 2
```

Como hemos comentado antes, el programa mediante un bucle que va desde el primer hasta el último tramo de torre, realiza los cálculos. Cada vez que realizamos un bucle, ponemos a cero ciertas variables, para que no haya problemas de cálculo, y además ejecutamos el procedimiento `Esfuerzo_Unitario`. Con ello calculamos los esfuerzos unitarios de cada tramo, ya que estos esfuerzos dependen de la geometría de cada tramo de torre.

Una vez calculados los esfuerzos unitarios se realizan los cálculos antes mencionados y asignamos los valores conseguidos a su correspondiente tabla.

```
For i = 1 To Dat(1)
  j = i * NInc(1)

  'Puesta a cero de las Variables
  For l = 1 To 6
    EsfUnitMont(l) = 0
    MOM2Caso1(l) = 0
    MOM2Caso2(l) = 0
  Next
  For l = 1 To 4
    EsfUnitDI(l) = 0
    EsfUnitDS(l) = 0
  Next

  For l = 1 To 9
    T(l, 1) = 0
    T(l, 2) = 0
  Next

  Call Esfuerzo_Unitario(i, EsfUnitMont, EsfUnitDS, EsfUnitDI)

  ki = i * NInc(1)
```

'CÁLCULO CASO 1-DIRECCIÓN HORIZONTAL

```

aux(1) = MomentosSecunCaso1(ki, 1) + MomentosSecunCaso1(ki, 2) - Abs(MomentosSecunCaso1(ki, 3)) + MomentosSecunCaso1(ki,
4)
T(1, 1) = i
T(2, 1) = 0 'Altura
T(3, 1) = (Dat(2) + Dat(3) + Peso(ki)) * Dat(10) 'Peso - Para los dos casos el mismo
T(4, 1) = FVientoElementos(1) + FVientoElementos(2) + EsfCortanteVientoCaso1(ki) 'Fuerza Cortante
T(5, 1) = MomentosCaso1(ki, 1) + MomentosCaso1(ki, 2) - MomentosCaso1(ki, 3) + MomentosCaso1(ki, 4) + aux(1)
T(6, 1) = FlechasCaso1(ki, 1) + FlechasCaso1(ki, 2) - FlechasCaso1(ki, 3) + FlechasCaso1(ki, 4)
T(7, 1) = T(5, 1) * EsfUnitMont(1) + T(3, 1) * EsfUnitMont(3) 'Tensión Montante
T(8, 1) = T(4, 1) * EsfUnitDS(1) 'Tensión Dia Sup
T(9, 1) = T(4, 1) * EsfUnitDI(1) 'Tensión Dia Inf
  
```

'CÁLCULO CASO 2-DIRECCIÓN DIAGONAL

```

aux(2) = MomentosSecunCaso2(ki, 1) + MomentosSecunCaso2(ki, 2) - Abs(MomentosSecunCaso2(ki, 3)) + MomentosSecunCaso2(ki,
4)
T(1, 2) = i
T(2, 2) = 0 'Altura
T(3, 2) = (Dat(2) + Dat(3) + Peso(ki)) * Dat(10) 'Peso - Para los dos casos el mismo
T(4, 2) = FVientoElementos(1) + FVientoElementos(2) + EsfCortanteVientoCaso1(ki) 'Fuerza Cortante
T(5, 2) = MomentosCaso2(ki, 1) + MomentosCaso2(ki, 2) - MomentosCaso2(ki, 3) + MomentosCaso2(ki, 4) + aux(2)
T(6, 2) = FlechasCaso2(ki, 1) + FlechasCaso2(ki, 2) - FlechasCaso2(ki, 3) + FlechasCaso2(ki, 4)
T(7, 2) = T(5, 2) * EsfUnitMont(2) + T(3, 2) * EsfUnitMont(3) 'Tensión Montante
T(8, 2) = T(4, 2) * EsfUnitDS(2) 'Tensión Dia Sup
T(9, 2) = T(4, 2) * EsfUnitDI(2) 'Tensión Dia Inf

For l = 1 To 9
    frmcalculotorre.Grid1(1).TextMatrix(i + 1, l) = T(l, 1)
    frmcalculotorre.Grid1(2).TextMatrix(i + 1, l) = T(l, 2)
Next

Next
  
```

Rutina Esfuerzo_Unitario. Los cálculos de esta rutina se han explicado antes. Debemos de distinguir que tipo de tramo se esta estudiando y para ello utilizamos la función If.

```

Dim Alfa As Double
Dim Beta As Double
  
```

```

'Los esfuerzos en los montantes son comunes tanto para el tramo Tipo 1 y Tipo 2
'Los esfuerzos en las diagonales en cambio son distintos
  
```

```

'Esfuerzos en el montante por flexión pura en dirección transversal a la sección. Caso 1
EsfUnitMont(1) = 1 / (2 * Tramo(i, 4) * Tramo(i, 6))
  
```


'Esfuerzos en el montante por flexión pura en dirección diagonal a la sección. Caso 2

$$\text{EsfUnitMont}(2) = 1 / (\text{Sqr}(2) * \text{Tramo}(i, 4) * \text{Tramo}(i, 6))$$

'Esfuerzos en el montante debido al peso propio de la estructura, teniendo en cuenta el peso de la góndola y aspas.

$$\text{EsfUnitMont}(3) = 1 / (4 * \text{Tramo}(i, 6))$$

'Si el tramo es de Tipo 1

If Tramo(i, 10) = 1 Then

'Ángulos de las Diagonales

$$\text{Alfa} = \text{Atn}(2 * \text{Tramo}(i, 3) / (\text{Tramo}(i, 2) * \text{Tramo}(i, 4)))$$

'Esfuerzos en la diagonal superior e inferior por esfuerzos cortante en dirección transversal a la sección. Caso 1

$$\text{EsfUnitDS}(1) = 1 / (4 * \text{Tramo}(i, 7))$$

$$\text{EsfUnitDI}(1) = 1 / (4 * \text{Cos}(\text{Alfa}) * \text{Tramo}(i, 8))$$

'Esfuerzos en la diagonal superior e inferior por esfuerzos cortante en dirección diagonal a la sección. Caso 2

$$\text{EsfUnitDS}(2) = 1 / (4 * \text{Sqr}(2) * \text{Tramo}(i, 7))$$

$$\text{EsfUnitDI}(2) = 1 / (4 * \text{Sqr}(2) * \text{Cos}(\text{Alfa}) * \text{Tramo}(i, 8))$$

Else

'Si el tramo es de Tipo 2

'Ángulos de las Diagonales

$$\text{Alfa} = \text{Atn}(\text{Tramo}(i, 11) / \text{Tramo}(i, 4))$$

$$\text{Beta} = \text{Atn}((\text{Tramo}(i, 3) / \text{Tramo}(i, 2) - \text{Tramo}(i, 11)) / \text{Tramo}(i, 4))$$

'Esfuerzos en la diagonal superior e inferior por esfuerzos cortante en dirección transversal a la sección. Caso 1

$$\text{EsfUnitDS}(1) = 1 / (2 * \text{Cos}(\text{Alfa}) * \text{Tramo}(i, 7))$$

$$\text{EsfUnitDI}(1) = 1 / (2 * \text{Cos}(\text{Beta}) * \text{Tramo}(i, 8))$$

'Esfuerzos en la diagonal superior e inferior por esfuerzos cortante en dirección diagonal a la sección. Caso 2

$$\text{EsfUnitDS}(2) = 1 / (2 * \text{Sqr}(2) * \text{Cos}(\text{Alfa}) * \text{Tramo}(i, 7))$$

$$\text{EsfUnitDI}(2) = 1 / (2 * \text{Sqr}(2) * \text{Cos}(\text{Beta}) * \text{Tramo}(i, 8))$$

End If

End Sub

10 CÁLCULOS VARIOS EN LA BASE

En este apartado, realizamos diversos cálculos de la base. Reacciones en la propia base, momentos de vuelco, reacciones horizontales...

Primero calculamos las reacciones en la base.

Como hemos comentado en el apartado anterior “9 Cálculo de esfuerzos en cada tramo de torre, montantes y diagonales”, los esfuerzos en los montantes son aquellos que contrarrestan los momentos flectores de flexión pura. De manera que para calcular las reacciones en la base, para los dos cálculos de estudio que estamos realizando, primero deberemos de calcular los momentos de vuelco que tenemos en la base, teniendo en cuenta también los momentos secundarios.

En este apartado utilizamos nuevas variables. Las explicamos brevemente

Peso Guardaremos el valor del peso de la estructura, teniendo en cuenta el peso de la góndola y aspás.

Altura Guardaremos el valor de la altura de la torre.

RPeso Guardaremos las reacciones en los montantes debido al peso de la estructura

RMomentoVuelco(2) Guardaremos las reacciones en los montantes debido a los momentos flectores, de los dos casos de estudio. De ahí que sea una matriz con 2 espacios.

MomentoTotal(2) Guardaremos los momentos flectores de vuelco de los dos casos de estudio. De ahí que sea una matriz con 2 espacios. Se utilizará también en los cálculos de cimentación.

EsfCortante(2) Guardaremos los esfuerzos cortantes de los dos casos de estudio. De ahí que sea una matriz con 2 espacios. Se utilizará también en los cálculos de cimentación.

Resultados(30) Guardaremos en esta variable los distintos resultados que obtenemos.

Lo primero que hacemos es poner estas variables a 0, para que no haya ningún tipo de error al realizar los cálculos.

Para calcular tanto el peso como la altura de la estructura utilizamos un bucle *For Next* desde el primer hasta el último tramo de torre.

$$Altura = Altura + Tramo(i, 3)$$

$$Peso = Peso + Tramo(i, 5)$$

Tramo(i, 3) Altura de tramo.

Tramo(i, 5) Peso de tramo.

Debemos de tener en cuenta también el peso de la góndola y aspás.

$$Peso = Peso + Dat(2) + Dat(3)$$

Siendo:

Dat(2) Peso de la góndola

Dat(3) Peso de las aspas.

Con el cálculo de los esfuerzos cortante y momento de vuelco utilizamos también un bucle *For Next*, pero esta vez desde el primer hasta el último incremento de torre. Para calcular el momento de vuelco simplemente multiplicaremos la fuerza ejercida por el viento en cada incremento por la altura del incremento respecto del suelo. Tanto la fuerza del viento ejercida en cada incremento como su altura respecto del suelo se han calculado en el apartado “5 Cálculo de la fuerza de viento ejercida en cada incremento”. Para calcular el esfuerzo cortante basta con sumar la fuerza ejercida por el viento en cada incremento.

Estos cálculos los realizamos para los dos casos de estudio. Pondremos como ejemplo sólo para el caso 1.

$$\begin{aligned} Esfcortante(1) &= Esfcortante(1) + Fviento(i) \\ MomentoTotal(1) &= MomentoTotal(1) + Fviento(i) \cdot AltViento(i) \end{aligned}$$

Al *MomentoTotal(1)* deberemos de sumarle también el momento causado por el peso descentrado de la góndola y aspas, y también los momentos secundarios. Este último ha sido calculado en el apartado “8 Cálculo de los momentos secundarios causados por las distintas sollicitaciones”.

$$MomentoTotal(1) = MomentoTotal(1) + MomentoSecundarioTotal(1) - MomentoPeso$$

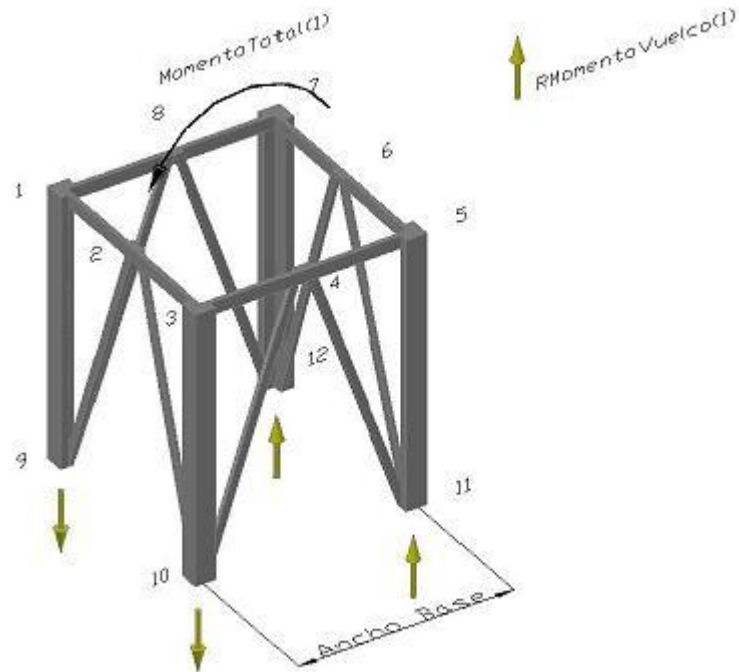
Una vez calculado el momento total calcularemos las reacciones en la base, primero para el Caso 1 y después para el Caso 2.

Las reacciones en la base debido al peso se dividen entre los cuatro montantes con lo que tendremos:

$$RPeso = \frac{Peso}{4}$$

A continuación calcularemos las reacciones debidas al momento de vuelco.

10.1 CASO 1. DIRECCIÓN PARALELO



$$\sum \text{Momento} = \text{MomentoTotal}(1) = 4 \cdot \text{RMomentoVuelco}(1) \cdot x$$

$$\text{RMomentoVuelco}(1) = \frac{\text{MomentoTotal}(1)}{4 \cdot x}$$

$$\text{RMomentoVuelco}(1) = \frac{a}{4 \cdot \frac{a}{2}}$$

$$\text{RMomentoVuelco}(1) = \frac{\text{MomentoTotal}(1)}{4 \cdot \frac{a}{2}} = \frac{\text{MomentoTotal}(1)}{2 \cdot a}$$

$$a = \text{Ancho}$$

Sumaremos las reacciones, pero debemos de tener en cuenta si las reacciones en los montantes están a compresión o a tracción. Si nos fijamos en las figuras tendremos.

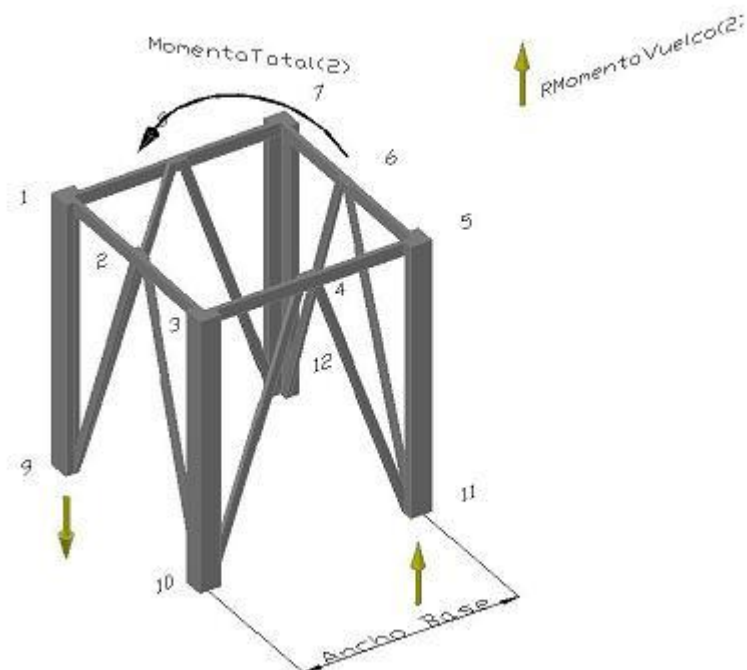
Reacciones en 11 y 12

$$\text{Resultados}(1) = R\text{Peso} + R\text{MomentoVuelco}(1)$$

Reacciones en 9 y 10

$$\text{Resultados}(2) = R\text{Peso} - R\text{MomentoVuelco}(1)$$

10.2 CASO 2. DIRECCIÓN DIAGONAL



$$\sum \text{Momentos} = \text{MomentoVuelco}(2) = 2 \cdot R\text{MomentoVuelco}(2) \cdot x$$

$$R\text{MomentoVuelco}(2) = \frac{\text{MomentoVuelco}(2)}{2 \cdot x}$$

$$\text{Siendo, } x = \sqrt{\left(\frac{a}{2}\right)^2 + \left(\frac{a}{2}\right)^2} = \sqrt{\frac{a^2}{4} + \frac{a^2}{4}} = \sqrt{\frac{2a^2}{4}} = a \sqrt{\frac{2}{4}} = a \sqrt{\frac{1}{2}} = \frac{a}{\sqrt{2}}$$

$$RMomentoVuelco(2) = \frac{MomentoVuelco(2)}{2 \cdot \frac{a}{\sqrt{2}}} = \frac{MomentoVuelco(2) \cdot \sqrt{2}}{2 \cdot a} =$$

$$\frac{MomentoVuelco(2) \cdot \sqrt{2}}{2 \cdot a} \cdot \frac{\sqrt{2}}{\sqrt{2}} = \frac{MomentoVuelco(2) \cdot 2}{2 \cdot a \cdot \sqrt{2}} = \frac{MomentoVuelco(2)}{a \cdot \sqrt{2}}$$

Sumaremos las reacciones, pero debemos de tener en cuenta si las reacciones en los montantes están a compresión o a tracción. Si nos fijamos en las figuras tendremos.

Reacciones en 11

$$Resultados(3) = RPeso + RMomentoVuelco(2)$$

Reacciones en 10 y 12

$$Resultados(4) = RPeso$$

Reacciones en 9

$$Resultados(5) = RPeso - RMomentoVuelco(2)$$

Como podemos observar las reacciones máximas siempre las tendremos en el apoyo 11 de la base.

10.3 EXPLICACIÓN RUTINA.

Declaración de variables.

Dim i As Integer

Dim j As Integer

Dim Peso As Double

Dim Altura As Double

Dim RPeso As Double

Dim RMomentoVuelco(2) As Double

Al comienzo de la rutina ponemos todas las variables a cero, para que no haya ningún problema al realizar los cálculos. *Esf cortante(2)* y *MomentoTotal(2)* al ser

matrices utilizamos un bucle *For Next* para poner sus valores a cero. Con *Peso* y *Altura* hacemos lo mismo.

'Puesta a 0 de las variables

```
For i = 1 To 2
    EsfCortante(i) = 0
    MomentoTotal(i) = 0
Next

Peso = 0
Altura = 0
```

Calculamos el peso y altura de la estructura. Utilizamos *For Next* desde el primer hasta el último tramo de torre. Una vez terminado el bucle debemos de tener en cuenta que al peso de la estructura debemos de sumarle el peso de la góndola y el peso de las aspas.

'Cálculo del peso y altura de la estructura

```
For i = 1 To Dat(1)
    Altura = Altura + Tramo(i, 3)
    Peso = Peso + Tramo(i, 5)
Next

Peso = Peso + Dat(2) + Dat(3)
```

Calculamos por fin el esfuerzo cortante de la base y el momento de vuelco, para los dos casos de estudio. Bucle que empieza desde el primer incremento hasta el último. Terminado el bucle al momento total de los dos casos sumamos los momentos secundarios y el momento causado por el peso de la góndola y aspas.

```
For i = 1 To NInc(2)
    EsfCortante(1) = EsfCortante(1) + FVientoCaso1(i)      'Dirección paralelo
    MomentoTotal(1) = MomentoTotal(1) + FVientoCaso1(i) * AltViento(i) 'Dirección paralelo
    EsfCortante(2) = EsfCortante(2) + FVientoCaso1(i)      'Dirección diagonal
    MomentoTotal(2) = MomentoTotal(2) + FVientoCaso1(i) * AltViento(i) 'Dirección diagonal
Next

MomentoTotal(1) = MomentoTotal(1) + MomentoSecundarioTotal(1) - MomentoPeso
MomentoTotal(2) = MomentoTotal(2) + MomentoSecundarioTotal(2) - MomentoPeso
```

Calculamos las reacciones como se ha explicado antes, y asignamos los resultados obtenidos a la variable *Resultados(30)*, y estos a su vez a sus correspondientes cajas de texto del formulario.

'Cálculo de las fuerzas en los montantes debido al peso de la estructura

$$RPeso = Peso / 4$$

'Caso 1

'Cálculo de las fuerzas en los montantes debido al momento flector

$$RMomentoVuelco(1) = MomentoTotal(1) / (Ancho * 2)$$

'A continuación las sumaremos teniendo en cuenta si son de compresión o tracción

$$Resultados(1) = RPeso - RMomentoVuelco(1) \quad \text{'Reacciones en 3 y 4}$$

$$Resultados(2) = RPeso + RMomentoVuelco(1) \quad \text{'Reacciones en 1 y 2}$$

'Caso 2

'Cálculo de las fuerzas en los montantes debido al momento flector

$$RMomentoVuelco(2) = MomentoTotal(2) / (Ancho * Sqr(2))$$

'A continuación las sumaremos teniendo en cuenta si son de compresión o tracción

$$Resultados(3) = RPeso + RMomentoVuelco(2) \quad \text{'Reacciones en 1}$$

$$Resultados(4) = RPeso \quad \text{'Reacciones en 2 y 3}$$

$$Resultados(5) = RPeso - RMomentoVuelco(2) \quad \text{'Reacciones en 4}$$

'Momentos de Vuelco

$$Resultados(6) = MomentoTotal(1) \quad \text{'Dirección Horizontal. Caso 1}$$

$$Resultados(7) = MomentoTotal(2) \quad \text{'Dirección Diagonal. Caso 2}$$

'Esfuerzos Cortantes

$$Resultados(8) = EsfCortante(1) \quad \text{'Dirección Horizontal. Caso 1}$$

$$Resultados(9) = EsfCortante(2) \quad \text{'Dirección Diagonal. Caso 2}$$

'Reacciones Máximas

$$Resultados(10) = Resultados(1) \quad \text{'Dirección Horizontal. Caso 1}$$

$$Resultados(11) = Resultados(3) \quad \text{'Dirección Diagonal. Caso 2}$$

'Peso Estructura

$$Resultados(12) = Peso$$

'Altura Estructura

$$Resultados(13) = Altura$$

'Asignaremos los resultados obtenidos a cada cuadro de texto en el formulario de resultados

Reacciones Caso 1

frmcalculobase.Text1(1) = Resultados(1)

frmcalculobase.Text1(2) = Resultados(2)

frmcalculobase.Text1(3) = Resultados(1)

frmcalculobase.Text1(4) = Resultados(2)

Reacciones Caso 2

frmcalculobase.Text1(5) = Resultados(3)

frmcalculobase.Text1(6) = Resultados(4)

frmcalculobase.Text1(7) = Resultados(4)

frmcalculobase.Text1(8) = Resultados(5)

Momentos, Reacciones Horizontales, Reacciones Máximas

frmcalculobase.Text1(9) = Resultados(6)

frmcalculobase.Text1(10) = Resultados(7)

frmcalculobase.Text1(11) = Resultados(8)

frmcalculobase.Text1(12) = Resultados(9)

frmcalculobase.Text1(13) = Resultados(10)

frmcalculobase.Text1(14) = Resultados(11)

frmcalculobase.Label3(0) = Ancho

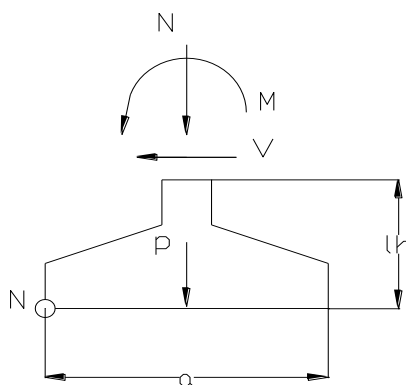
11 CÁLCULOS DE CIMENTACIÓN

Para el cálculo de la cimentación se debe considerar la hipótesis más desfavorable que se haya obtenido a partir de los resultados anteriores. Con las reacciones que se generan en la base se deben efectuar los cálculos de la cimentación.

Dada la forma de la estructura, se pueden pensar varios tipos de cimentación. Si la base es de gran tamaño, se podría efectuar una cimentación independiente de cada montante, esto podría ser recomendable si el terreno es de calidad y se pueden garantizar que los asentamientos y corrimientos se producirán de una manera conjunta en los cuatro montantes. Sin embargo, siempre se tiene la posibilidad de hacer solo una zapata que cubra los cuatro montantes, para así poder garantizar la uniformidad del comportamiento de la estructura.

11.1 COMPROBACIÓN AL VUELCO Y AL DESLIZAMIENTO.

Para efectuar los cálculos de la cimentación se han seguido las indicaciones de la Norma EHE y de los apuntes de la asignatura “Cálculo de Estructuras” de I.T.I Mecánica. La primera comprobación que debe efectuarse en zapatas sometidas a momentos o fuerzas horizontales de alguna importancia es la seguridad al vuelco. La condición correspondiente se obtiene expresando que los momentos estabilizadores de las fuerzas exteriores, respecto al punto A de la figura supera a los momentos de vuelco:



$$(N + P) \cdot \frac{a}{2} \geq (M + V \cdot l_h) \cdot \gamma_1$$

Con los siguientes significados:

N, M, V : esfuerzo normal, momento flector y esfuerzo cortante en cara superior de cimentación.

P = peso propio de la zapata.

a = ancho de la zapata.

l_h = altura de la zapata.

γ_1 = coeficiente de seguridad al vuelco, para el que suele tomarse 1,5.

11.2 DISTRIBUCIÓN DE PRESIONES DEL TERRENO.

La distribución de las presiones del terreno sobre la base de una zapata (que interesa para comprobar que no se rebasa la presión admisible de éste y para calcular los esfuerzos en la zapata), depende fundamentalmente del tipo de suelo y de la rigidez de la zapata. En nuestro estudio suponemos una distribución de presiones uniforme. Con lo que presión obtenida no debe de superar a la presión admisible del terreno, es decir, debe de cumplirse:

$$\sigma = \frac{N + P}{a} \leq \sigma_{adm}$$

Donde:

N es la carga centrada de servicio por unidad de longitud de muro.

P es el peso propio de la zapata.

a es el ancho de la zapata.

Estudiamos dos casos posibles que dependerán de la excentricidad. Con lo que primero debemos de conseguir la excentricidad resultante en la base de la zapata.

$$e = \frac{M_{Total}}{V_{Total}} = \frac{M + V \cdot l_h}{N + P}$$

1.- Caso de carga actuando con una excentricidad reducida $e \leq a/6$ (resultante dentro del núcleo central) se obtiene una distribución trapecial. La presión máxima en el borde de la zapata vale:

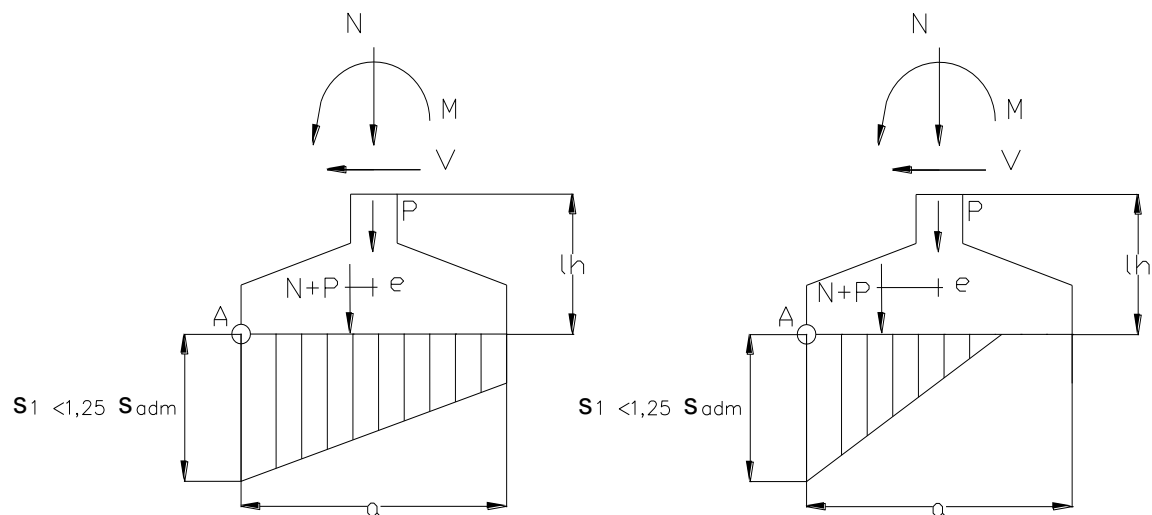
$$\sigma_1 = \frac{N + P}{a} \left(1 + \frac{6e}{a} \right)$$

Debe de verificarse que:

$$\sigma_1 \leq 1,25 \cdot \sigma_{adm}$$

2.- Caso de carga actuando con una excentricidad elevada $e > a/6$ (resultante fuera del núcleo central) se obtiene una distribución triangular, pues no es posible que se produzcan tracciones. En este caso, la presión máxima en el borde de la zapata vale:

$$\sigma_1 = \frac{4}{3} \cdot \frac{N + P}{(a - 2 \cdot e)}$$



Todas estas ecuaciones e ideas son las que aplicaremos para realizar nuestro estudio de zapatas.

11.3 CÁLCULOS.

Primero, se debe efectuar un cálculo a vuelco de la zapata, pero para ello primero se debe fijar un espesor de la zapata, ese dato se introduce como altura de la zapata en la introducción de datos de cimentación.

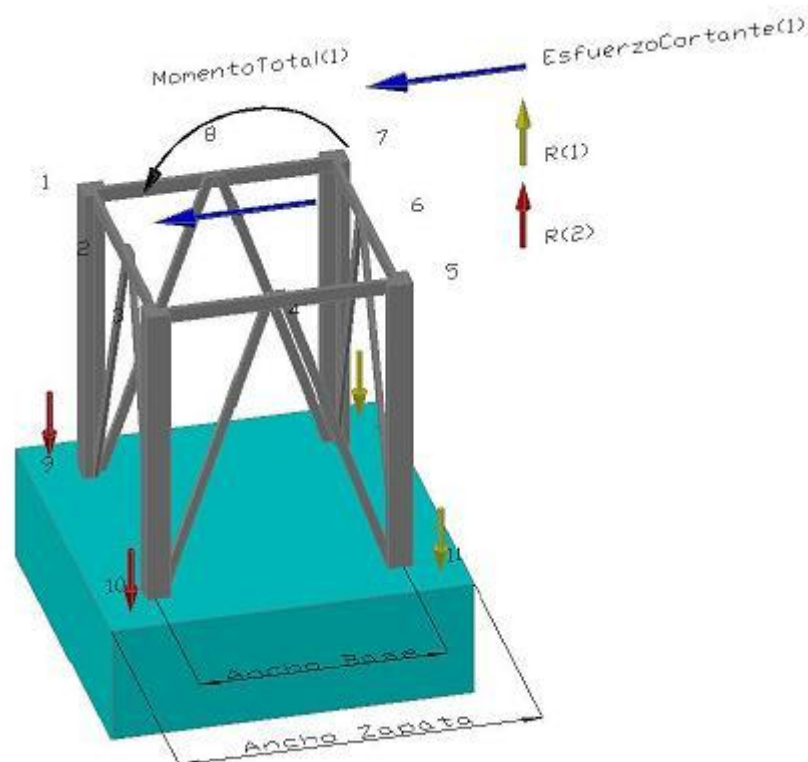
Como hasta ahora se han calculado dos posibles situaciones de trabajo del aerogenerador, calcularemos las dimensiones de la zapata para los dos casos, y nos quedaremos con el más desfavorable. A continuación se calculará el cálculo a vuelco de la zapata de los dos casos.

Aplicando los conocimientos adquiridos en la asignatura “Cálculo de Estructuras”, sobre dimensionamiento de zapatas y lo anteriormente explicado sabemos que:

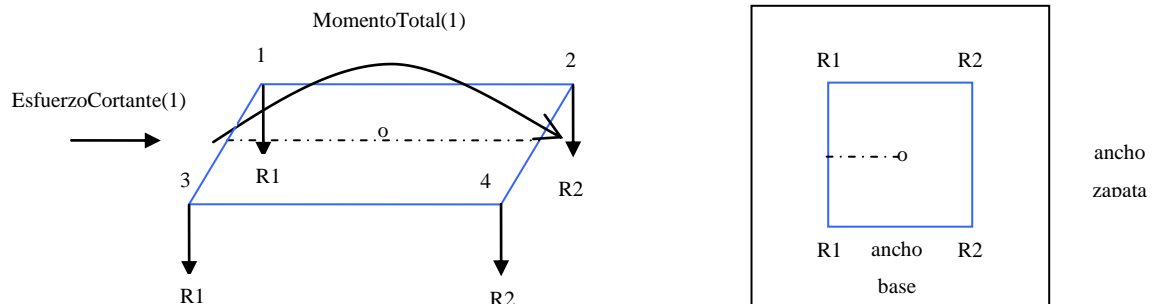
$$\gamma_f \geq \frac{\sum \text{Momentos Estabilizadores}}{\sum \text{Momento Vuelco}}$$

Los Momentos Estabilizadores los componen las reacciones en la base y el peso de la zapata. Los momentos de vuelco son el generado por los esfuerzos cortantes y el momento total que tenemos en la base, teniendo en cuenta los momentos secundarios. γ_f es el coeficiente de seguridad al vuelco. Se suele tomar un valor de 1.5.

Con la siguiente figura queda todo mucho más claro.



Dirección Paralelo. Caso 1



$$(N + P) \cdot \frac{a}{2} \geq (M + V \cdot l_h) \cdot \gamma_1$$

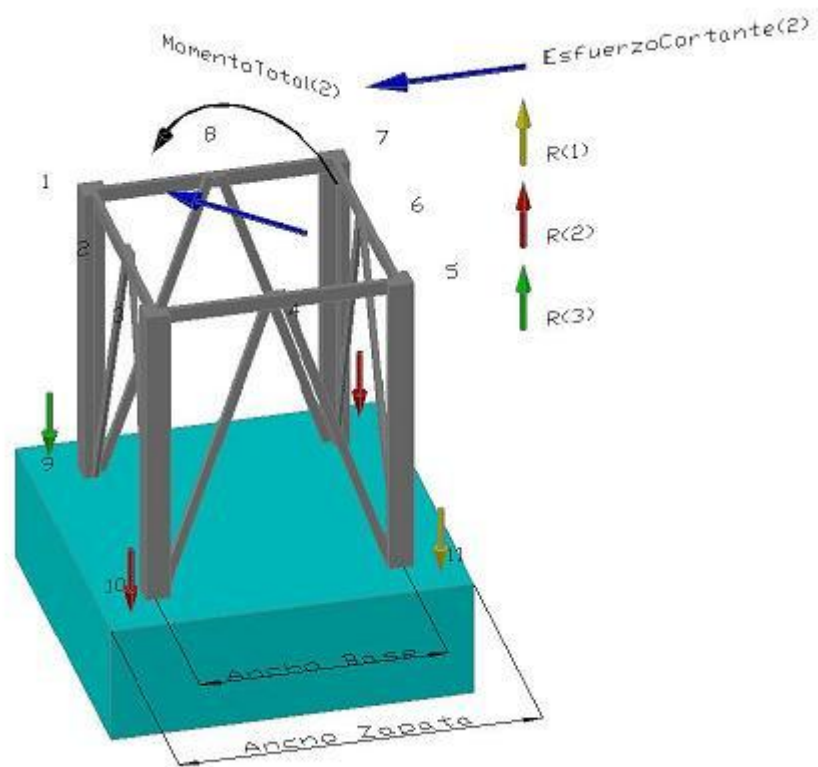
Si aplicamos la anterior ecuación tenemos que:

$$1.5 \geq \frac{10 \cdot \rho \cdot a^2 \cdot h \cdot \frac{a}{2} \cdot 10 + 2 \cdot R_1 \cdot \left(\frac{a}{2} + \frac{\text{ancho}}{2}\right) + 2 \cdot R_2 \cdot \left(\frac{a}{2} - \frac{\text{ancho}}{2}\right)}{M + V \cdot h}$$

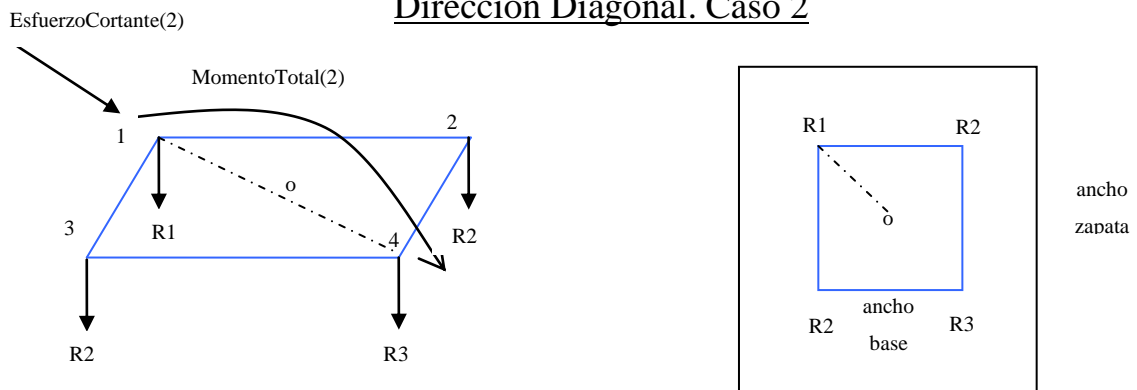
$$1.5 \cdot (M + V \cdot h) = \frac{10 \cdot \rho \cdot a^3 \cdot h}{2} + a \cdot (R_1 + R_2) + \text{ancho} \cdot (R_1 - R_2)$$

$$\frac{10 \cdot \rho \cdot a^3 \cdot h}{2} + a \cdot (R_1 + R_2) + \text{ancho} \cdot (R_1 - R_2) - 1.5 \cdot (M + V \cdot h)$$

Obtenemos una ecuación de tercer grado, que luego resolveremos.



Dirección Diagonal. Caso 2



Igual que con el Caso 1, aplicamos la ecuación de seguridad al vuelco.

$$1.5 \geq \frac{\rho \cdot a^2 \cdot h \cdot \frac{a}{2} \cdot 10 + (R_1 + R_2) \cdot \left(\frac{a}{2} + \frac{\text{ancho}}{2}\right) + (R_2 + R_3) \cdot \left(\frac{a}{2} - \frac{\text{ancho}}{2}\right)}{M + V \cdot h}$$

$$1.5 \cdot (M + V \cdot h) = \frac{10 \cdot \rho \cdot a^3 \cdot h}{2} + \frac{R_1}{2} \cdot (a + \text{ancho}) + R_2 \cdot a + \frac{R_3}{2} \cdot (a - \text{ancho})$$

$$\frac{10 \cdot \rho \cdot a^3 \cdot h}{2} + a \cdot \left(\frac{R_1}{2} + R_2 + \frac{R_3}{2} \right) + ancho \cdot \left(\frac{R_1}{2} - \frac{R_3}{2} \right) - 1.5 \cdot (M + V \cdot h)$$

Tenemos dos ecuaciones de tercer grado. El programa Visual Basic, no tiene ninguna función del tipo FindRoot, como en programas tipo MatLab o Mathematica, de manera que creamos un pequeño bucle, donde intentaremos hallar el valor de la anchura de la zapata para los dos casos de estudio. Nos quedaremos por supuesto con el valor más alto.

Lo que haremos será darle valores a la anchura de la zapata empezando desde cero hasta un máximo de 30 metros de anchura. El valor de la anchura irá aumentando de centímetro en centímetro. Como sabemos el valor de esas ecuaciones de tercer grado debe de ser mayor que cero, de manera que en el momento que sea mayor que cero y menor que uno se saldrá del bucle y tendremos ya nuestro valor de la anchura de la zapata.

Este bucle lo realizaremos con los dos casos de estudio. Nos quedamos con el valor más alto, y a continuación realizamos el estudio de las tensiones y excentricidad comprobando la seguridad al deslizamiento.

Para comprobar la excentricidad, tomamos como fuerza axil la suma de todas las reacciones que encontramos de los cuatro montantes.

11.4 EXPLICACIÓN RUTINA.

Dim a(2) As Double
 Dim Densidad As Double
 Dim h As Single
 Dim Aux As Double
 Dim N As Double 'Axil Centrado
 Dim ex As Double 'Excentricidad
 Dim Tension As Double 'Tensión Máxima
 Dim reaccion As Double

Densidad = Dat(11)

h = Dat(13)

For a(1) = 1 To 30 Step 0.01

Aux = (Densidad * 10 * h * a(1) ^ 3) / 2 + a(1) * (Resultados(1) + Resultados(2)) + Ancho * (Resultados(1) - Resultados(2)) - 1.5 * (MomentoTotal(1) + EsfCortante(1) * h)

If Aux > 0 Then

If Aux < 0 Then

If Aux > 1 Then

'Resultados(3) = a(1)

GoTo sigue1

End If

End If

End If

Next

sigue1:

Aux = 0

For a(2) = 1 To 30 Step 0.01

Aux = (Densidad * h * a(2) ^ 3) / 2 + a(2) * (Resultados(3) / 2 + Resultados(4) + Resultados(5) / 2) + Ancho * (Resultados(3) / 2 - Resultados(5) / 2) - 1.5 * (MomentoTotal(2) + EsfCortante(2) * h)

If Aux > 0 Then

If Aux < 1 Then

'Resultados(3) = a(1)

GoTo sigue2

Javier Vergara

```

    End If

    End If

Next

sigue2:

If a(1) > a(2) Then

N = Resultados(1) + Resultados(2) + Resultados(1) + Resultados(2)

Else

N = Resultados(3) + Resultados(4) + Resultados(4) + Resultados(5)

a(1) = a(2)

End If


ex = (MomentoTotal(1) + EsfCortante(1) * h) / (N + Densidad * h * a(1) ^ 3)

If ex > a(1) / 6 Then

'Distribución Triangular

Tension = (N + Densidad * h * a(1) ^ 3) / a(1) ^ 2 * (1 + 6 * e / a(1))

Resultados(15) = "Triangular"

Else

'Distribución Trapecial

Tension = 4 / 3 * (N + Densidad * h * a(1) ^ 3) / ((a(1) - 2 * e) * a(1))

Resultados(15) = "Trapecial"

End If


Resultados(14) = Tension

MsgBox ("El valor mínimo de la zapata es " & FormatNumber(a(1), 0))


'Asignaremos los resultados obtenidos a cada cuadro de texto en el formulario de resultados

frmcalculobase.Text1(15) = CLng(a(1))

frmcalculobase.Text1(16) = h

```

Javier Vergara

frmcalculobase.Text1(17) = CLng(Resultados(14))

frmcalculobase.Text1(18) = Resultados(15)

frmcalculobase.Label3(2) = CLng(a(1))

frmcalculobase.Label3(4) = CLng(a(1))

frmcalculobase.Label3(3) = h

End Sub

12 SALIDA DE RESULTADOS

Después de haber explicado todos los cálculos, vemos que resultados hemos obtenido. Tenemos dos formularios en los que hemos diferenciado por un lado los resultados propios de la torre, y por otro lado los resultados que obtenemos en la base y el pequeño estudio de cimentación.

Empezando por los resultados propios de la torre. Vemos como los resultados obtenidos son para cada tramo de torre, pero en cada tramo hay varios incrementos, de manera que hemos tomado los valores del incremento más bajo de todos ellos, ya que estos son los más desfavorables.

Los resultados de tramos de torre son los siguientes:

- Altura (m): Es la altura, respecto del suelo, a la que se encuentra la base del tramo.
- Peso (Kg): Es la suma de los pesos por encima del tramo, teniendo en cuenta el peso de la góndola y el peso de las aspas
- Esfuerzo Cortante (N): Es la fuerza que ejerce el viento sobre el último incremento del tramo, teniendo en cuenta todos los esfuerzos cortantes que tiene por encima.
- Momento (N·m): Es la suma de todos los momentos que aparecen en el último incremento del tramo. Es decir la suma del momento debido al viento, al peso descentrado de la góndola y aspas, momentos secundarios.
- Flecha (m): Deformación que ha sufrido el tramo de torre debido a todas las solicitaciones
- Tensión en el Montante (N/m²).
- Tensión Celosía Superior (N/m²).
- Tensión Celosía Inferior (N/m²).

Todos estos resultados se obtienen para los dos casos de estudio que se han realizado a lo largo del programa.

En el siguiente formulario, hemos introducido los cálculos referidos al cálculo de la base y al pequeño estudio de cimentación.

Cálculos de la base

Se calculan las reacciones en los cuatro montantes de la base, es decir en los montantes del último tramo de torre. Todas estas reacciones están calculadas en Kg. Tenemos también el momento de vuelco que se da en toda la estructura. Este momento es la suma de todos los momentos. Momento causado por el viento, por el peso descentrado del peso de

la góndola y aspas, y momentos secundarios. Por último la reacción máxima que se da en los montantes del último tramo.

Cálculos de cimentación.

Al realizar estos cálculos lo que se intenta es dimensionar la zapata de manera aproximada. Una vez dimensionada se obtiene la tensión máxima y el tipo de distribución de tensión que tiene.



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

***“Cálculo de aerogenerador en celosía mediante
software basado en Visual Basic”***

MANUAL PROGRAMADOR

Javier Vergara Lecue

Tutores: Cesar Díaz de Cerio

Pamplona, 24 de Noviembre de 2010

Índice

Página

PARTE 1: INTRODUCCIÓN Y CONCEPTOS TEÓRICOS..... 10

1 INTRODUCCIÓN	11
1.1 Modo de diseño y modo de ejecución.....	11
1.2 Formularios y controles	11
1.3 Objetos y propiedades	12
1.3.1 Nombres de Objetos	13
1.3.2 Propiedades de Objetos	13
1.4 Eventos.....	14
1.4.1 Formularios	14
1.4.2 Cuadros de Texto y Grids(Tablas)	15
1.4.3 Botones, Barras de Menú y Herramientas.....	15
1.5 Métodos.....	16
1.6 Proyectos y ficheros	16
1.7 Variables	17
1.8 Sentencias de controles	18
1.8.1 Sentencia IF...THEN...ELSE.....	19
1.8.2 Sentencia FOR...NEXT	19
1.8.3 Sentencia WHILE WEND	20
1.8.4 Sentencia SELECT...CASE	20
1.8.5 Sentencia DO...LOOP	21

1.9 Funciones para manejo de caracteres	21
---	----

PARTE 2: PROGRAMA 23

1 INTRODUCCIÓN	24
----------------------	----

2 PROPIEDADES FORMULARIOS	25
---------------------------------	----

2.1 Formulario datos	25
----------------------------	----

2.1.1 Formulario	25
------------------------	----

2.1.2 Etiquetas	25
-----------------------	----

2.1.3 Cajas de texto	27
----------------------------	----

2.1.4 Combo	29
-------------------	----

2.1.5 Flexgrid. Tablas.....	29
-----------------------------	----

2.1.3 Botones.....	29
--------------------	----

2.1.4 Barra de herramientas.....	30
----------------------------------	----

2.1.5 Barra de menú	30
---------------------------	----

2.2 Formulario cálculo torre.....	31
-----------------------------------	----

2.2.1 Formulario	31
------------------------	----

2.2.2 Etiquetas	31
-----------------------	----

2.2.3 Flexgrid. Tablas.....	31
-----------------------------	----

2.2.4 Botones	32
---------------------	----

2.2.5 Barra de herramientas.....	32
----------------------------------	----

2.2.6 Barra de menú	32
---------------------------	----

2.3 Formulario cálculo base	33
2.3.1 Formulario	33
2.3.2 Etiquetas	33
2.3.3 Cajas de texto	35
2.3.4 Botones	37
2.3.5 Barra de herramientas	37
2.3.6 Barra de menú	38
2.4 Formulario hoja guardar	38
2.4.1 Formulario	38
2.4.2 Cajas de texto	38
2.4.3 Barra de herramientas	39
2.4.4 Barra de menú	39
2.4 Formulario guardar como	39
2.5.1 Formulario	39
2.5.2 Botones	39
3 ALGORÍTMOS FORMULARIOS	40
3.1 Formulario principal. FRMDATOS	40
3.1.1 Formulario	40
3.1.1.1 Form_Load	40
3.1.1.2 Form_Activate	41
3.1.1.3 Form_OLEDragDrop	41
3.1.1.4 Form_Unload	42
3.1.2 Cajas de texto	44

3.1.2.1 Text1_KeyPress	44
3.1.2.2 Text1_Click.....	45
3.1.2.3 Text2_KeyPress	46
3.1.3 Grids. Tablas	46
3.1.3.1 Grid1_KeyPress	46
3.1.3.2 Grid1_KeyUp	46
3.1.4 Botones.....	47
3.1.4.1 cmdbajar_Click	47
3.1.4.2 cmdsubir_Click	47
3.1.4.3 cmdeliminar_Click	48
3.1.4.4 cmdlimpiardatos_Click	49
3.1.4.5 cmdlimpiartramos_Click.....	50
3.1.4.6 cmdinsertar_Click	50
3.1.4.7 cmdcalculotorre_Click	51
3.1.4.8 cmdcalculobase_Click.....	52
3.1.5 Botones menú.....	53
3.1.5.1 mnuabrir_Click	53
3.1.5.2 mnuabrirbase_Click	54
3.1.5.3 mnuguardartexto_Click.....	55
3.1.5.4 mnuguardarexcel_Click	56
3.1.5.5 mnuimprimir_Click.....	56
3.1.5.6 mnusalir_Click	56
3.1.5.7 mnuedicioncortar_Click	57

3.1.5.8 mnuedicioncopiar_Click	57
3.1.5.9 mnuedicionpegar_Click	57
3.1.6 Botones barra de herramientas	58
3.2 Formulario cálculo de la torre. FRMCALCULOTORRE	59
3.2.1 Formulario	59
3.2.1.1 Form_Load	59
3.2.2 Botones	60
3.2.2.1 cmdexcel_Click	60
3.2.2.2 cmdtexto_Click	61
3.2.3 Botones menú	61
3.2.3.1 mnuguardartexto_Click	61
3.2.3.2 mnuguardarexcel_Click	61
3.2.3.3 mnuimprimir_Click	62
3.2.3.4 mnusalir_Click	62
3.2.3.5 mnuedicioncortar_Click	62
3.2.3.6 mnuedicioncopiar_Click	62
3.2.4 Botones barra de herramientas	63
3.3 Formulario cálculo de la base. FRMCALCULOBASE	63
3.3.1 Botones menú	63
3.3.1.1 mnuguardartexto_Click	63
3.3.1.2 mnuguardarexcel_Click	64
3.3.1.3 mnuimprimir_Click	64
3.3.1.4 mnusalir_Click	64

3.3.1.5 mnuedicioncortar_Click	64
3.3.1.6 mnuedicioncopiar_Click	64
3.3.1.9 mnuedicionpegar_Click	65
3.3.2 Botones barra de herramientas	65
3.4 Formulario hoja guardar. FRMHOJAIMPRIMIR	66
3.4.1 Botones menú	66
3.4.1.1 mnuguardartexto_Click	66
3.4.1.2 mnuimprimir_Click	66
3.4.1.3 mnusalir_Click	67
3.4.1.4 mnuedicioncortar_Click	67
3.4.1.5 mnuedicioncopiar_Click	67
3.4.1.6 mnuedicionpegar_Click	67
3.4.1.7 mnuedicionselecciontodo_Click	67
3.4.2 Botones barra de herramientas	68
3.5 Formulario guardar como. FRMGUARDAR	69
3.5.1 Excel	69
3.5.2 Texto	69
4 MÓDULOS	70
4.1 Módulo Funciones	70
4.1.1 AbrirExcel	71
4.1.2 AbrirTexto	74
4.1.3 InsertarDatosTorre	79
4.1.4 AbrirBase	82

4.2 Módulo GuardarDatos.....	83
------------------------------	----

***PARTE 3: MANUAL* 86**

1 FORMULARIO PRINCIPAL	87
------------------------------	----

2 FORMULARIO CÁLCULO TORRE	90
----------------------------------	----

3 FORMULARIO CÁLCULOBASE	92
--------------------------------	----

4 FORMULARIO GUARDAR	93
----------------------------	----

PARTE 1

INTRODUCCIÓN Y CONCEPTOS TEÓRICOS

1. INTRODUCCIÓN

Visual Basic es uno de los lenguajes de programación que más entusiasmo despiertan entre los programadores de PCs, tanto expertos como novatos. En el caso de programadores expertos por la facilidad con la que se desarrollan aplicaciones complejas en poquísimos minutos. En el caso de programadores novatos por el hecho de ver lo que son capaces a los pocos minutos de empezar su aprendizaje. El precio que hay que pagar por utilizar Visual Basic es una menor velocidad o eficiencia en las aplicaciones.

Visual Basic es un lenguaje de programación visual. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

1.1 MODO DE DISEÑO Y MODO DE EJECUCIÓN

La aplicación Visual Basic de Microsoft puede trabajar de dos modos distintos: en modo de diseño y en modo de ejecución. En modo de diseño el usuario construye interactivamente la aplicación, colocando controles en el formulario, definiendo sus propiedades, y desarrollando funciones para gestionar los eventos.

La aplicación se prueba en modo de ejecución. En ese caso el usuario actúa sobre el programa (introduce eventos) y prueba como responde el programa. Hay algunas propiedades de los controles que deben de establecerse en modo de diseño, pero muchas otras pueden cambiarse en tiempo de ejecución desde el programa escrito en Visual Basic. También hay propiedades que sólo pueden establecerse en modo de ejecución y que no son visibles en modo de diseño.

1.2 FORMULARIOS Y CONTROLES

Cada uno de los elementos gráficos que pueden formar parte de una aplicación típica de Windows es un tipo de control: los botones, las cajas de diálogo y de texto, las cajas de selección desplegables, los botones de opción y de selección, las barras de desplazamiento horizontales y verticales, los gráficos, los menús, y muchos otros tipos de

elementos son controles para Visual Basic. Cada control debe de tener un nombre a través del cual se puede hacer referencia a él en el programa. Visual Basic proporciona nombres por defecto que el usuario puede modificar.

En la terminología de Visual Basic se llama formulario (form) a una ventana. Un formulario puede ser considerado como una especie de contenedor para los controles. Una aplicación puede tener varios formularios, pero un único formulario puede ser suficiente para las aplicaciones más sencillas. Los formularios deben también tener un nombre, que puede crearse siguiendo las mismas reglas que para los controles.

1.3 OBJETOS Y PROPIEDAS

Los formularios y los distintos tipos de controles son entidades genéricas de las que puede haber varios ejemplares concretos en cada programa. En programación orientada a objetos se llama clase a estas entidades genéricas, mientras que se llama objeto a cada ejemplar de una clase determinada. Por ejemplo, en un programa puede haber varios botones, cada uno de los cuales es un objeto del tipo de control command button, que sería la clase.

Cada formulario y cada tipo de control tienen un conjunto de propiedades que definen su aspecto gráfico (tamaño, color, posición en la ventana, tipo y tamaño de la letra, etc.) y su forma de responder a las acciones del usuario (si está activo o no, por ejemplo.) Cada propiedad tiene un nombre que viene ya definido por el lenguaje.

Por lo general, las propiedades de un objeto son datos que tienen valores lógicos (True, False) o numéricos concretos, propios de ese objeto y distintos de las de otros objetos de su clase. Así pues, cada clase, tipo de objeto o control tiene su conjunto de propiedades, y cada objeto o control concreto tiene unos valores determinados para las propiedades de su clase.

Casi todas las propiedades de los objetos pueden establecerse en tiempo de diseño y también, casi siempre, en el tiempo de ejecución. En este segundo caso se accede a sus valores por medio de las sentencias del programa, en forma análoga a como se accede a cualquier variable en un lenguaje de programación. Para ciertas propiedades ésta es la única forma de acceder a ellas. Por supuesto Visual Basic permite crear distintos tipos de variables.

Se puede acceder a una propiedad de un objeto por medio del nombre del objeto a que pertenece, seguido de un punto y el nombre de la propiedad, como por ejemplo `optColor.objName`.

1.3.1 Nombres de objetos

En principio cada objeto de Visual Basic debe tener un nombre, por medio del cual se hace referencia a dicho objeto. El nombre puede ser el que el usuario desee, e incluso Visual Basic proporciona nombres por defecto para los diversos controles. Estos nombres por defecto hacen referencia al tipo de control y van seguidos de un número que se incrementa a medida que se van introduciendo más controles de ese tipo en el formulario (por ejemplo VScroll, para una barra de desplazamiento –scroll bar-vertical, HScroll, para una barra horizontal, etc.)

1.3.2 Propiedades de objetos

En un principio explicaremos cada una de las propiedades que podemos encontrar de todos los objetos. No se explican todas las propiedades, si no sólo aquellas que hemos cambiado para realizar el programa. Las demás se dejan por defecto.

(Nombre): Devuelve el nombre usado en el código para identificar un objeto.

Alignment: Devuelve o establece la alineación del texto de un control.

BackColor: Devuelve o establece el color de fondo usado para mostrar texto y gráficos en un objeto.

BorderStyle: Devuelve o establece el estilo del borde de un objeto.

Caption: Devuelve o establece el texto mostrado en la barra de título de un objeto o bajo el icono de un objeto.

Cols: Determina el número total de columnas en un FlexGrid.

Enabled: Devuelve o establece un valor que determina si un objeto puede responder a eventos generados por el usuario.

FillColor: Devuelve o establece el color usado para rellenar formas, círculos y cuadrados.

FixedCols: Devuelve o establece el número total de columnas fijas (no desplazables) de un Flexgrid.

FixedRows: Devuelve o establece el número total de filas fijas (no desplazables) de un Flexgrid.

Font: Establece el tipo de letra, tamaño, subrayado, tachado... del objeto.

ForeColor: Devuelve o establece el color del primer plano usado para mostrar textos y gráficos en un objeto.

Height: Devuelve o establece el alto de un objeto.

Index: Devuelve o establece el número que identifica un control en una matriz de controles.

Left: Devuelve o establece la distancia entre el borde izquierdo interno de un objeto y el

borde izquierdo de su contenedor.

MergeCells: Devuelve o establece si celdas con el mismo contenido se deben agrupar en una celda única que contiene múltiples filas o columnas. Para el Flexgrid.

OLEDropMode: Devuelve o establece si este objeto puede actuar como destino para una operación colocar de OLE. Su función es determinar si se pueden arrastrar archivos al objeto.

Rows: Devuelve el número total de filas en un FlexGrid.

StartPosition: Devuelve o establece un valor que especifica la posición de un formulario cuando aparece por primera vez.

TabIndex: Devuelve o establece el orden de tabulación de un objeto en su formulario primario.

Text: Devuelve o establece el texto contenido en el control. Sólo para las cajas de texto.

ToolTipText: Devuelve o establece el texto mostrado cuando el mouse se sitúa sobre un control.

Top: Devuelve o establece la distancia entre el borde superior interno de un objeto y el borde superior de su contenedor.

Width: Devuelve o establece el ancho de un objeto.

1.4 EVENTOS

Se ha dicho que las acciones del usuario sobre el programa se llaman eventos. Son eventos típicos el clicar sobre un botón, el hacer doble clic sobre el nombre de un fichero para abrirlo, el arrastrar un icono, el pulsar una tecla o combinación de teclas, el elegir una opción de un menú, el escribir en una caja de texto, o simplemente mover el ratón.

Cada vez que se produce un evento sobre un determinado tipo de control, Visual Basic arranca una determinada función o procedimiento que realiza la acción programada por el usuario para ese evento concreto. Estos procedimientos se llaman con un nombre que se forma a partir del nombre del objeto y el nombre del evento, separados por el carácter (_), como por ejemplo TextBox_Click, que es el nombre del procedimiento que se ocupará de responder al evento click en el objeto TextBox.

Los objetos de los que está compuesto el programa, aparte de tener unas propiedades, ejecutarán una serie de algoritmos, según el evento al que estén expuestos. A algoritmos llamamos a todos los procedimientos y funciones que el programa debe ejecutar, para que no haya problemas y pueda llevarse a cabo eficientemente.

Como se ha comentado antes, cada uno de los objetos posibles, formularios y controles, dispone de una determinada serie de eventos a los que puede responder.

Explicaremos en un pequeño resumen el tipo de eventos que se han utilizado en el programa, para cada uno de los objetos.

1.4.1 Formularios

Load: El evento Load se activa al cargar el formulario. Con el formulario principal esto sucede al arrancar la ejecución de un programa; con el resto de los formulario al mandarlos cargar desde cualquier procedimiento o al hacer referencia a alguna propiedad o control de un formulario que no esté cargado. El evento Load de un formulario se puede utilizar para ejecutar una función que dé valor a sus propiedades y a las de los controles que dependen de dicho formulario. No se puede utilizar para dibujar o imprimir sobre el formulario, pues en el momento en que se produce este evento el formulario todavía no está disponible para dichas operaciones.

Unload: Al descargar un formulario se produce el evento Unload. Para pasar por el evento Unload es necesario cerrar la ventana con el botón cerrar o llamarlo explícitamente.

Activate: Cuando un formulario pasa a ser la ventana activa se produce el evento Activate.

Hide: Podemos ocultar un formulario sin descargarlo con el método Hide. Esto hace que el formulario desaparezca de la ventana, aunque sus variables y propiedades sigan estando accesibles y conservando sus valores.

Show: Para hacer visible un formulario oculto pero ya cargado se utiliza el método Show. Si el formulario no había sido cargado previamente, el método Show genera el evento Load.

DragOver: El evento DragDrop sucede al concluir una operación de arrastrar y soltar.

1.4.2 Cuadros de Texto y Grids (Tablas)

KeyPress: El evento KeyPress sucede cuando el usuario pulsa y suelta determinada tecla. En el procedimiento asociado con este evento el único argumento KeyAscii es necesario para conocer cuál es el código ASCII de la tecla pulsada.

KeyUp: El evento KeyUp se produce cuando el usuario suelta una tecla.

1.4.3 Botones. Barra de Menú. Barra de Herramientas.

Click: El evento Click se activa cuando el usuario pulsa y suelta rápidamente uno de los botones del ratón. También ocurre al pulsar un botón con la tecla Intro o la barra espaciadora sobre el control.

Con este pequeño resumen de los eventos que encontramos en el programa, realizaremos una pequeña descripción en cada formulario, de todos los eventos que podemos encontrar

1.5 MÉTODOS

Los métodos son funciones que también son llamadas desde programa, pero a diferencia de los procedimientos no son programados por el usuario, sino que vienen ya pre-programadas con el lenguaje. Los métodos realizan tareas típicas, previsibles y comunes para todas las aplicaciones. De ahí que vengan con el lenguaje y que se libere al usuario de la tarea de programarlos. Cada tipo de objeto o de control tiene sus propios métodos.

Por ejemplo, los controles gráficos tienen un método llamado Line que se encarga de dibujar líneas rectas. De la misma forma existe un método llamado Circle que dibuja circunferencias y arcos de circunferencia. Es obvio que el dibujar líneas rectas o circunferencias es una tarea común para todos los programadores y que Visual Basic da ya resuelta.

1.6 PROYECTOS Y FICHEROS

Cada aplicación que se empieza a desarrollar en Visual Basic es un nuevo proyecto. Un proyecto comprende otras componentes más sencillas, como por ejemplo los formularios (que son ventanas de la interface de usuario de la nueva aplicación) y los módulos (que son conjuntos de funciones y procedimientos sin interface gráfica de usuario).

Un proyecto se compone siempre de varios ficheros (al menos de dos) y hay que preocuparse de guardar cada uno de ellos en el directorio adecuado. Existe siempre un fichero con extensión *.vbp (Visual Basic Project) que se crea con el comando File/Save Project As. El fichero del proyecto contiene toda la información de conjunto. Además hay que crear un fichero por cada formulario y por cada módulo que tenga el proyecto. Los ficheros de los formularios se crean con File/Save Filename As teniendo como extensión *.frm. Los ficheros de código o módulos se guardan también con el comando File/Save

Filename As y tiene como extensión *.bas si se trata de un módulo estándar o *.cls si se trata de un módulo de clase.

Clicando en el botón Save en la barra de herramientas se actualizan todos los ficheros del proyecto. Si no se habían guardado todavía en el disco, Visual Basic abre cajas de diálogo Save As por cada uno de los ficheros que hay que guardar.

1.7 VARIABLES

Una variable es un nombre que designa a una zona de memoria (se trata por tanto de un identificador), que contiene un valor de tipo información. Tal y como su nombre indica, las variables pueden cambiar su valor a lo largo de la ejecución de un programa. Completando a las variables existe lo que se denomina constantes las cuales son identificadores pero con la particularidad de que el valor que se encuentra en ese lugar de la memoria sólo puede ser asignado una única vez. El tratamiento y tipos de datos es igual al de las variables.

Para poder identificar las variables y poder utilizarlas en el programa tenemos las distintas opciones del siguiente cuadro.

Tipo de variable	Lugar de declaración	Accesibilidad
Global o Public	Declaraciones de *.bas	Desde todos los formularios
Dim o Private	Declaraciones de *.bas	Desde todas las funciones de ese módulo
Public	Declaraciones de *.frm	Desde cualquier procedimiento del propio formulario y desde otros precedida del nombre del módulo en el que se ha declarado
Dim o Private	Declaraciones de *.frm	Desde cualquier procedimiento del propio formulario
Dim	Cualquier procedimiento de un módulo	Desde el propio procedimiento

Al igual que C y otros lenguajes de programación, Visual Basic dispone de distintos tipos de datos, aplicables tanto para constantes como para variables. La siguiente tabla muestra los tipos de datos disponibles en Visual Basic.

Tipo	Descripción	Carácter de declaración	Rango
Boolean	Binario		True o False
Byte	Entero corto		0 a 255
Integer	Entero (2 bytes)	%	-32768 a 32767
Long	Entero largo (4 bytes)	&	-2147483648 a 2147483647
Single	Real simple precisión (4 bytes)	!	-3.40E+38 a 3.40E+38
Double	Real doble precisión (8 bytes)	#	-1.79D+308 a 1.79D+308
Currency	Número con punto decimal fijo (8 bytes)	@	-9.22E+14 a 9.22E+14
String	Cadena de caracteres (4 bytes + 1 byte/car hasta 64K)	\$	0 a 65500 caracteres
Date	Fecha (8 bytes)		1 de enero de 100 a 31 de diciembre de 9999. Indica también la hora
Variant	Fecha/hora, números enteros, reales o caracteres	Ninguno	F/h: como Date números: mismo rango que el tipo de valor almacenado
User-defined	Cualquier tipo de dato o estructura de datos. Se crean utilizando la sentencia Type	ninguno	

1.8 SENTENCIAS DE CONTROL

Las sentencias de control, denominadas estructuras de control, permiten tomar decisiones y realizar un proceso repetidas veces. Son los denominados bifurcaciones y bucles. Este tipo de estructuras son comunes en cuanto a concepto en la mayoría de los lenguajes de programación, aunque sus sintaxis puede variar de un lenguaje de programación a otro. Se trata de unas estructuras muy importantes ya que son las encargadas de controlar el flujo de un programa según los requerimientos del mismo. En el programa se utilizan las siguientes estructuras de control:

- **If...Then...Else**
- **For...Next**
- **While...Wend**
- **Select Case**
- **Do...Loop**

1.8.1 Sentencia *IF...THEN...ELSE*

Esta estructura permite ejecutar condicionalmente una o más sentencias y puede escribirse de dos formas. La primera sólo ocupa una línea y tiene la forma siguiente:

If condición **Then** sentencial1 [**Else** Sentencial2]

La segunda es más general y se muestra a continuación:

If condición **Then**

Sentencia(s)

[**Else**

Sentencia(s)]

End if

Si *condición* es *True (verdadera)*, se ejecutan las sentencias que están a continuación de *Then*, y si condición es *False (falsa)*, se ejecutan las sentencias que están a continuación de *Else*, si esta cláusula ha sido especificada, pues es opcional.

1.8.2 Sentencia *FOR...NEXT*

La sentencia For da lugar a un lazo o bucle, y permite ejecutar un conjunto de sentencias cierto número de veces. Su forma general es:

For variable = **expresion1** **To** **expresion2** [**Step** **expresion3**]
Sentencia(s)

Next [variable]

Cuando se ejecuta una sentencia *For*, primero se asigna el valor de la *expresion1* a la variable y se comprueba si su valor es mayor o menor que la *expresion2*. En caso de ser menor se ejecutan las sentencias, y en caso de ser mayor el control del programa salta a las líneas a continuación de *Next*. Todo esto sucede en caso de ser la *expresion3* positiva. En caso contrario se ejecutarán las sentencias cuando la variable sea mayor que *expresion2*. Una vez ejecutadas las sentencias, la variable se incrementa en el valor de la *expresion3*, o en 1 si *Step* no se especifica, volviéndose a ejecutar la comparación entre la variable y la *expresion2*, y así sucesivamente.

1.8.3 Sentencia **WHILE...WEND**

Esta sentencia es otra forma de generar bucles que se recorren mientras se cumpla la condición inicial. Su estructura es la siguiente:

```
While condición
    Sentencia(s)
Wend
```

Por ejemplo,

Counts = 0	‘Se inicia la variable
While Counts < 20	‘Se comprueba el valor de Counts
Counts = Counts + 1	‘Se incrementa el valor de Counts
Wend	‘Se acaba el bucle cuando Counts > 19

1.8.4 Sentencia **SELECT CASE**

Esta sentencia permite ejecutar una de entre varias acciones en función del valor de una expresión. Es una alternativa a **If...Then...ElseIf** cuando se compara la misma expresión con diferentes valores. Su forma general es la siguiente:

```
Select Case expresión
Case etiqueta1
    Sentencia(s)
Case etiqueta2
    Sentencia(s)
Case Else
    Sentencias(s)
End Select
```

donde **expresion** es una expresión numérica o alfanumérica, y **etiqueta1**, **etiqueta2**, ... pueden adoptar las formas siguientes:

- *expresion*
- *expresion To expresion*
- **Is** *operador-de-relación expresion*
- *combinación de las anteriores separadas por comas*

1.8.5 Sentencia *DO...LOOP*

Un **Loop** (*bucle*) repite la ejecución de un conjunto de sentencias mientras una condición dada sea cierta, o hasta que una condición dada sea cierta. La condición puede ser verificada antes o después de ejecutarse el conjunto de sentencias. La forma utilizada en el programa es la siguiente.

Do [{While} condicion]

Sentencia(s)

Loop

1.9 FUNCIONES PARA MANEJO DE CADENAS DE CARACTERES

Existen varias funciones útiles para el manejo de cadenas de caracteres (Strings). Estas funciones se utilizan para la evaluación, manipulación o conversión de cadenas de caracteres. Las utilizadas en el programa se muestran en la siguiente tabla.

Función en Visual Basic	Utilidad	Comentarios
Len(string varname)	Número de caracteres de una cadena	
InStr([n], string1, string2)	Devuelve un entero que especifica la posición inicial de la primera aparición de una cadena dentro de otra.	Devuelve la posición de string2 en string1, a partir del parámetro n que es opcional
InStrRev(string1, string2)	Devuelve la posición de la primera aparición de una cadena dentro de otra, comenzando por el extremo derecho de la cadena.	Devuelve la posición de string2 en string1
Left(string, length)	Devuelve una cadena que contiene un número especificado de caracteres a partir del lado izquierdo de una cadena.	
Mid(string, ini [,n])	Devuelve una cadena de caracteres, con un número especificado de caracteres de una cadena.	El parámetro n es opcional e indica el número de caracteres a extraer a partir de "ini"
Replace(string, substring, replacewith)	Buscar y reemplazar una subcadena por otra en una cadena	Reemplaza substring por replacewith
IsNumeric(string)	Devuelve un valor Boolean que indica si una expresión puede evaluarse como un número.	
Space(number)	Devuelve una cadena que	

Javier Vergara

	consta del número especificado de espacios.	
--	--	--

PARTE 2

PROGRAMA

1 INTRODUCCIÓN.

La función principal del programa es la de realizar los cálculos oportunos. Pero como todo programa que se precie hoy en día, debe de tener opciones tales como Abrir, Guardar, Guardar Como... Estas opciones también se explican en este apartado. También se detalla las propiedades de todos los elementos que forman el programa.

El programa está compuesto por 4 formularios. Explicamos todos ellos detenidamente, describiendo las propiedades de todos los objetos que hay en ellos. Nos referimos como objetos a las cajas de texto, etiquetas... Describimos también todos los algoritmos. Explicaremos y haremos una lista de todos los objetos y algoritmos que hay en cada uno de los formularios.

2 PROPIEDADES FORMULARIOS

Se describen en tablas todas las propiedades de los objetos. Se explicarán de formulario en formulario

2.1 FORMULARIO DE DATOS

15 Etiquetas.
 14 Cajas de Texto.
 1 Combo.
 2 Flexgrid. Cuadros.
 8 Botones.
 1 Barra Herramientas.
 1 Barra de Menú.

Las propiedades de los objetos están descritas en tablas, para su fácil descripción.

2.1.1 Formulario

(Nombre)	Frmdatos
Caption	Aerogenerador
Height	8760
OLEDropMode	1-Manual
Widht	11700
WindowState	0-Normal

2.1.2 Etiquetas

(Nombre)	Label1	Label1	Label1
Caption	Número de Tramos:	Peso Góndola:	Peso Aspas:
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	255	255	255
Index	1	2	3
Left	720	720	720
Top	1080	1440	1800
Widht	1575	1575	1575

(Nombre)	Label1	Label1	Label1
Caption	Distancia CG Góndola:	Distancia hasta aspas:	Superficie Frontal Góndola:
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	255	255	255
Index	4	5	6
Left	720	720	720
Top	2160	2520	2880
Width	1815	1815	2055

(Nombre)	Label1	Label1	Label1
Caption	Altura Góndola:	Diámetro aspas:	Viento de Referencia:
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	255	255	255
Index	7	8	9
Left	720	720	720
Top	3240	3600	3960
Width	1215	1935	1935

(Nombre)	Label1	Label1	Label1
Caption	Coefficiente peso propio:	Densidad hormigón (Kg/m ³):	Tensión admisible terreno :
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	255	255	255
Index	10	11	12
Left	720	4920	4920
Top	4320	1080	1440
Width	1935	2055	2055

(Nombre)	Label1	Label1	Label1
Caption	Altura zapata (m):	Introducción del tipo de Tramo de Torre	Se introducen por orden empezando desde arriba
Font	Ms Sans Serif Normal 8	Ms Sans Serif Negrita 10	Ms Sans Serif Normal 8
Height	255	255	255
Index	13	14	15
Left	4920	4800	5040
Top	1800	2640	3000
Width	2055	4200	3495

(Nombre)	Label2	Label2	Label2
Caption	DATOS TORRE	DATOS CIMENTACIÓN	TRAMOS DE TORRE
Font	Times New Roman Normal 14	Times New Roman Normal 14	Times New Roman Normal 14
Height	255	255	255
Index	1	2	3
Left	1560	5040	4440
Top	600	600	4680
Width	2055	3135	2655

2.1.3 Cajas de Texto

(Nombre)	Text1	Text1	Text1
Alignment	1-Right Justify	1-Right Justify	1-Right Justify
Enabled	False	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	1	2	3
Left	3000	3000	3000
TabIndex	0	1	2
Top	960	1320	1680
Width	1155	1155	1155

(Nombre)	Text1	Text1	Text1
Alignment	1-Right Justify	1-Right Justify	1-Right Justify
Enabled	True	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	4	5	6
Left	3000	3000	3000
TabIndex	3	4	5
Top	2040	2400	2760
Width	1155	1155	1155

(Nombre)	Text1	Text1	Text1
Alignment	1-Right Justify	1-Right Justify	1-Right Justify
Enabled	True	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	7	8	9
Left	3000	3000	3000
TabIndex	6	7	8
Top	3120	3480	3840
Width	1155	1155	1155

(Nombre)	Text1	Text1	Text1
Alignment	1-Right Justify	1-Right Justify	1-Right Justify
Enabled	True	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	10	11	12
Left	3000	7200	7200
TabIndex	9	10	11
Top	4200	1080	1440
Width	1155	1155	1155
(Nombre)	Text1	Text2	
Alignment	1-Right Justify	2-Center	
Enabled	True	True	
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	
Height	360	360	
Index	13		
Left	7200	6600	
TabIndex	12	13	
Top	1800	3435	
Width	1155	1245	

2.1.4 Combo

(Nombre)	Combo
Font	Ms Sans Serif Normal 10
Height	360
Left	7200
TabIndex	14
Top	4800
Width	1815

2.1.5 Flexgrid. Tablas

(Nombre)	Grid1	Grid2
Alignment	Center	Center
BackColor	&H8000000D&	
BackColorSel	&H8000000D&	
Cols	11	11
FixedCols	0	0
FixedRows	1	1
Font	Courier New Normal 9	Courier New Normal 9
Height	2250	495
Left	120	120
MergeCells	0-flexMergeNerver	2-flexMergeRestricRows
Rows	10	2
TabIndex	24	25
Top	5520	5085
Width	11225	11225

2.1.6 Botones

(Nombre)	cmdcalculotorre	cmdcalculobase	cmdlimpiardatos
Caption	Cálculo Torre	Cálculo Base	Limpiar Datos
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	495	495	495
Left	8760	1080	9360
Top	960	960	1680
Width	1215	1095	1335

(Nombre)	cmdinsetar	cmdsubir	cmdbajar
Caption	Insertar	Subir	Bajar
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	495	495	495
Left	8040	4800	5880
Top	3360	3960	3960
Width	855	855	855

(Nombre)	cmdeliminar	cmdlimpiartramos
Caption	Eliminar	Limpiar Tramos
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	495	495
Left	6960	8040
Top	3960	3960
Widht	855	855

2.1.7 Barra de Herramientas

Botones:

- Abrir
- Guardar
- Imprimir
- Cortar
- Copiar
- Pegar
- Ayuda

2.1.8 Barra de Menú

Archivo:

- Abrir
- Abrir Base de Datos
- Guardar Como...
 - Archivo Texto
 - Archivo Excel
- Imprimir
- Salir

Edición:

- Cortar
- Copiar
- Pegar

Ayuda:

2.2 FORMULARIO CÁLCULO TORRE

- 2 Etiquetas.
- 4 Flexgrid. Cuadros.
- 2 Botones.
- 1 Barra Herramientas.
- 1 Barra de Menú.

2.2.1 Formulario

(Nombre)	frmcalculotorre
Caption	Aerogenerador - Cálculo Torre
Height	10740
Width	12210
WindowState	0-Normal

2.2.2 Etiquetas

(Nombre)	Label1	Label1
Caption	CASO 1- DIRECCIÓN HORIZONTAL	CASO 2- DIRECCIÓN DIAGONAL
Font	Times New Roman Normal 14	Times New Roman Normal 14
Height	255	255
Index	1	2
Left	4440	4500
Top	480	4800
Width	4575	4455

2.2.3 Flexgrid. Tablas

(Nombre)	Grid1	Grid1
Alignment	Center	Center
BackColor	&H8000000D&	&H8000000D&
BackColorSel	&H8000000D&	&H8000000D&
Cols	13	13
FixedCols	0	0
FixedRows	1	1
Font	Courier New Normal 9	Courier New Normal 9
Height	3600	3600
Index	1	2
Left	120	120
MergeCells	0-flexMergeNerver	0-flexMergeNerver
Rows	3	3
TabIndex	1	2
Top	1080	5400

Width	11730	11730
-------	-------	-------

(Nombre)	Grid2	Grid2
Alignment	Center	Center
Cols	13	13
FixedCols	0	0
FixedRows	1	1
Font	Courier New Normal 9	Courier New Normal 9
Height	495	495
Index	1	2
Left	120	120
MergeCells	2-flexMergeRestricRows	2-flexMergeRestricRows
Rows	2	2
Top	795	5115
Width	11730	11730

2.2.4 Botones

(Nombre)	cmdexcel	cmdtexto
Caption	Todo Excel	Todo Texto
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	615	615
Left	4800	6120
Top	9120	9120
Width	975	975

2.2.5 Barra de Herramientas

Botones:

- Guardar
- Imprimir
- Cortar
- Copiar
- Pegar

2.2.6 Barra de Menú

Archivo:

- Guardar Como...
- Archivo Texto
- Archivo Excel
- Imprimir
- Salir

Edición:

Javier Vergara

- Cortar
- Copiar
- Pegar

2.3 FORMULARIO CÁLCULOS BASE Y CIMENTACIÓN

- 31 Etiquetas.
- 18 Cajas de Texto.
- 2 Botones.
- 2 Cuadros de Imágenes.
- 1 Barra de Herramientas.
- 1 Barra de Menú.

2.3.1 Formulario

(Nombre)	frmcalculobase
Caption	Aerogenerador - Cálculo Base
Height	10860
Width	12075
WindowState	0-Normal

2.3.2 Etiquetas

(Nombre)	Label1	Label1	Label1
Caption	1	2	3
Font	Ms Sans Serif Negrita 10	Ms Sans Serif Negrita 10	Ms Sans Serif Negrita 10
Height	255	255	255
Index	1	2	3
Left	2040	3360	4680
Top	1560	1560	1560
Width	180	180	180

(Nombre)	Label1	Label1	Label1
Caption	4	CASO 1	CASO 2
Font	Ms Sans Serif Negrita 10	Times New Roman Normal 12	Times New Roman Normal 12
Height	255	255	255
Index	4	5	6
Left	6000	600	600
Top	1560	2160	2760
Width	180	900	900

(Nombre)	Label1	Label1	Label1
Caption	CASO 1	CASO 2	Par de Vuelco (N·m)
Font	Times New Roman Normal 12	Times New Roman Normal 12	Ms Sans Serif Normal 8
Height	255	255	255
Index	7	8	9
Left	1560	5160	240
Top	3600	3600	4200
Width	900	900	1575

(Nombre)	Label1	Label1	Label1
Caption	Reacción Horizontal (N)	Reacción Máxima (N)	Longitud Zapata (m)
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	255	255	255
Index	13	14	15
Left	3840	3840	1560
Top	4680	5160	7080
Width	1815	1695	1455

(Nombre)	Label1	Label1	Label1
Caption	Altura Zapata (m)	Tensión Zapata (Kg/m²)	Tipo Distribución de Tensiones
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	255	255	255
Index	16	17	18
Left	3840	3840	1560
Top	7560	8040	8520
Width	1215	1695	2415

(Nombre)	Label2	Label2	Label2
Caption	CÁLCULOS BASE	REACCIONES	CÁLCULOS CIMENTACIÓN
Font	Times New Roman Normal 14	Times New Roman Normal 12	Times New Roman Normal 14
Height	255	255	255
Index	1	2	3
Left	3000	3360	1560
Top	600	1080	6240
Width	2295	1575	3615

(Nombre)	Label3	Label3	Label3
Caption			
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	255	255	255
Index	1	2	3
Left	10800	6360	8280
Top	3120	9120	9120
Width	495	495	495

(Nombre)	Label3
Caption	
Font	Ms Sans Serif Normal 8
Height	255
Index	4
Left	10920
Top	8640
Width	495

(Nombre)	Label4	Label4	Label4
Caption	CASO 2	CASO 1	1
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Negrita 12
Height	255	255	255
Index	1	2	3
Left	7440	7440	8280
Top	1440	3120	1560
Width	615	615	180

(Nombre)	Label4	Label4	Label4
Caption	2	3	4
Font	Ms Sans Serif Negrita 12	Ms Sans Serif Negrita 12	Ms Sans Serif Negrita 12
Height	255	255	255
Index	4	5	6
Left	10560	8280	10560
Top	1560	4560	4560
Width	180	180	180

2.3.3 Cajas de Texto

(Nombre)	Text1	Text1	Text1
Alignment	2-Center	2-Center	2-Center
Enabled	True	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	1	2	3
Left	1560	2880	4200
TabIndex	0	1	2
Top	2040	2040	2040
Width	1095	1095	1095

(Nombre)	Text1	Text1	Text1
Alignment	2-Center	2-Center	2-Center
Enabled	True	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	4	5	6
Left	5520	1560	2880
TabIndex	3	4	5
Top	2040	2640	2640
Width	1095	1095	1095

(Nombre)	Text1	Text1	Text1
Alignment	2-Center	2-Center	1-Right Justify
Enabled	True	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	7	8	9
Left	4200	5520	2280
TabIndex	6	7	8
Top	2640	2640	4080
Width	1095	1095	1200

(Nombre)	Text1	Text1	Text1
Alignment	1-Right Justify	1-Right Justify	1-Right Justify
Enabled	True	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	10	11	12
Left	5880	2280	5880
TabIndex	9	10	11
Top	4080	4560	4560
Width	1200	1200	1200

(Nombre)	Text1	Text1	Text1
Alignment	1-Right Justify	1-Right Justify	1-Right Justify
Enabled	False	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	13	14	15
Left	2280	3000	3960
TabIndex	12	13	14
Top	5040	5040	6960
Width	1200	1200	1200

(Nombre)	Text1	Text1	Text1
Alignment	1-Right Justify	1-Right Justify	1-Right Justify
Enabled	True	True	True
Font	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10	Ms Sans Serif Normal 10
Height	360	360	360
Index	16	17	18
Left	3960	3960	3960
TabIndex	15	16	17
Top	7440	7920	8400
Width	1200	1200	1200

2.3.4 Botones

(Nombre)	cmdexcel	cmdtexto
Caption	Todo Excel	Todo Texto
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	615	615
Left	2160	3480
Top	9240	9240
Width	975	975

2.3.5 Barra de Herramientas

Botones:

- Guardar
- Imprimir
- Cortar
- Copiar
- Pegar

2.3.6 Barra de Menú

Archivo:

- Guardar Como...
- Archivo Texto
- Archivo Excel
- Imprimir
- Salir

Edición:

- Cortar
- Copiar
- Pegar

2.4 FORMULARIO HOJA DE IMPRIMIR/GUARDAR

1 Caja de Texto.

1 Barra de Herramientas.

1 Barra de Menú.

2.4.1 Formulario

(Nombre)	frmhojaimprimir
Caption	Hoja de Impresión
Height	10650
Widht	13155
WindowState	0-Normal

2.4.2 Cajas de Texto

(Nombre)	Text1
Alignment	0- Left Justify
Font	Courier New Normal 10
Height	9000
Left	240
Top	600
Widht	12500

2.4.3 Barra de herramientas

Botones:

- Guardar
- Imprimir
- Cortar
- Copiar
- Pegar

Barra de Menú

Archivo:

- Guardar
- Imprimir
- Salir

Edición:

- Cortar
- Copiar
- Pegar
- Seleccionar Todo

2.5 FORMULARIO GUARDAR COMO

3 Botones

2.5.1 Formulario

(Nombre)	frmguardar
Caption	Guardar Como...
Height	1995
Width	4785
WindowState	0-Normal

2.5.2 Botones

(Nombre)	cmdexcel	Cmdtexto	cmdcancelar
Caption	Excel	Texto	Cancelar
Font	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8	Ms Sans Serif Normal 8
Height	495	495	495
Left	600	1920	3240
Top	840	840	840
Width	855	855	855

3 ALGORÍTMOS FORMULARIOS

Una vez que hemos explicado todas las propiedades de todos los objetos de los formularios, explicaremos todos los algoritmos que encontramos dentro de estos formularios.

3.1 FORMULARIO PRINCIPAL. FRMDATOS

3.1.1 Formulario

3.1.1.1 Form_Load:

Con este evento cargamos el formulario principal. En el describimos las propiedades de los dos Grids (tablas). Sólo en modo de ejecución se puede describir ciertas propiedades de estas tablas. La anchura de columnas, textos que pueden aparecer en cada casilla, orientación de texto...

De manera que lo primero que se hace al cargar el formulario es describir las tablas. Utilizamos la función With, para hacer referencia a la tabla en cuestión, y luego simplemente aplicamos la propiedad elegida. Describimos el texto que debe aparecer en la casilla deseada y a continuación la anchura de las columnas. Con un bucle For Next, aplicamos a todas las columnas una alineación de texto central.

'Describimos las tablas

With Grid1

```
.TextArray(1) = "Tramo"  
.TextArray(2) = "Código"  
.TextArray(3) = "Pasos"  
.TextArray(4) = "m"  
.TextArray(5) = "m"  
.TextArray(6) = "Kg"  
.TextArray(7) = "Montante"  
.TextArray(8) = "Celosía Sup"  
.TextArray(9) = "Celosía Inf"  
.TextArray(10) = "Tramo m²"  
.TextArray(11) = "Tramo"  
.TextArray(11) = "Diagonal"  
.ColWidth(0) = 0  
.ColWidth(1) = 800  
.ColWidth(2) = 1100  
.ColWidth(3) = 800  
.ColWidth(4) = 900  
.ColWidth(5) = 950
```

```
.ColWidth(6) = 950
.ColWidth(7) = 1000
.ColWidth(8) = 1350
.ColWidth(9) = 1350
.ColWidth(10) = 1700
```

End With

With Grid2

```
.RowHeight(0) = 450
.TextArray(1) = "Nº"
.TextArray(3) = "Nº"
.TextArray(4) = "Altura"
.TextArray(5) = "Lg Ancho"
.TextArray(6) = "Peso"
.TextArray(7) = "Sección cm²"
.TextArray(8) = "Sección cm²"
.TextArray(9) = "Sección cm²"
.TextArray(10) = "Área Sup Viento"
.TextArray(11) = "Tipo"
.TextArray(12) = "Altura "
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1100
.ColWidth(3) = 800
.ColWidth(4) = 900
.ColWidth(5) = 950
.ColWidth(6) = 950
.ColWidth(7) = 1000
.ColWidth(8) = 1350
.ColWidth(9) = 1350
.ColWidth(10) = 1700
.ColWidth(12) = 1330
.MergeRow(0) = True
```

End With

'Damos la orientación de texto deseada a las columnas

For i = 1 To 12

```
Grid1.ColAlignment(i) = 4
Grid2.ColAlignment(i) = 4
```

Next

Descritas las tablas, debemos de abrir la base de datos de donde obtendremos los tramos de torre. En la misma carpeta del programa hay un fichero de texto. El fichero de texto se utiliza para ubicar la dirección de la base de datos. En la variable DirOriginal y RutaBaseDatos guardamos esta dirección y en la variable TablaOriginal y NombreTabla guardamos el nombre de la tabla que estamos abriendo de la base de datos. Utilizando esta dirección ubicamos la base de datos, y a continuación la abrimos. Una vez abierta pasamos los códigos de todos los tramos al Combo.

```
'Abrimos el texto y obtenemos la dirección de la base de datos
nFic = FreeFile
Open "BaseDatos.cfg" For Input As nFic
    Line Input #nFic, lineatexto
    Line Input #nFic, lineatexto
    i = InStr(lineatexto, Chr(34))
    j = InStr(i + 1, lineatexto, Chr(34))
    RutaBaseDatos = Mid(lineatexto, i + 1, j - i - 1)
    DirOriginal = RutaBaseDatos
Close

DirOriginal = DirOriginal + "\BaseDatos"
RutaBaseDatos = DirOriginal
TablaOriginal = "Aerogenerador"
NombreTabla = TablaOriginal

'Abrimos la base de datos de los tramos de torre
Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal)
Set RsTorre = DbBaseDatos.OpenRecordset(TablaOriginal, dbOpenDynaset)

'Llevamos todos los códigos de tramo al combo
RsTorre.MoveLast
RsTorre.MoveFirst
Combo.AddItem " "
For i = 1 To RsTorre.RecordCount
    Combo.AddItem RsTorre(0)
    RsTorre.MoveNext
Next
RsTorre.MoveFirst

'Al índice de las cajas de texto le asignamos el valor 2
xi = 2
```

3.1.1.2Form_Activate

Utilizamos este evento para escribir junto al nombre del formulario “Aerogenerador”, la dirección de la base de datos que se esté utilizando en ese momento.

```
frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos
```

3.1.1.3 Form_OLEDragDrop

Utilizamos este evento para arrastras archivos de texto o de Excel, en los cuales hayamos guardado anteriormente configuraciones de torre, para poder cargar los datos. Confirmamos primeramente que es un archivo de datos lo que se arrastra y que no se ha arrastrado más de un archivo a la vez. Si ocurre esto se sale del evento.

```
'Si no es un archivo lo que se arrastra, habrá error  
If Data.GetFormat(vbCFFiles) = False Then  
    MsgBox "No se ha arrastrado ningún archivo", vbInformation  
    Exit Sub  
End If  
  
'Si se arrastra más de un archivo, habrá error  
If Data.Files.Count > 1 Then  
    MsgBox "No se puede arrastrar más de un archivo"  
    Exit Sub  
End If
```

Comprobado esto borramos todos los datos de formularios. En la variable RutaFichero guardamos la dirección del archivo para poder luego abrirlo. Vemos si el archivo es de Excel o de texto, para poder abrir el archivo con la función adecuada.

```
cmdlimpiardatos_Click  
cmdlimpiartramos_Click  
  
Dim posnombre As Integer  
Dim Texto As String  
  
RutaFichero = Data.Files(1)  
posnombre = InStrRev(RutaFichero, ".")  
Texto = Mid(RutaFichero, posnombre + 1)
```

```
'Comprobamos si es un archivo Excel o de texto
If Texto = "txt" Then
    Call AbrirTexto
Else
    Call AbrirExcel
End If

final:
```

3.1.1.4Form_Unload

Con este evento simplemente cerramos el programa.

```
End
```

3.1.2 Cajas de Texto

3.1.2.1 Text1_KeyPress

En los textos sólo podremos introducir números. Al pulsar una tecla en el texto si no es un número aparece un mensaje de advertencia diciendo que sólo está permitido introducir números. Para que no haya problemas de cálculo, los decimales se distinguen por medio del “.”. Si pulsamos la “,” esta cambia automáticamente al “.”.

```
Dim X As Integer

'Sólo se pueden teclear en los textos número y "."
If KeyAscii <> Asc("9") Then
If KeyAscii <> Asc("8") Then
If KeyAscii <> Asc("7") Then
If KeyAscii <> Asc("6") Then
If KeyAscii <> Asc("5") Then
If KeyAscii <> Asc("4") Then
If KeyAscii <> Asc("3") Then
If KeyAscii <> Asc("2") Then
If KeyAscii <> Asc("1") Then
If KeyAscii <> Asc("0") Then
If KeyAscii <> 8 Then      'tecla suprimir
    X = 1
End If
End If
End If
End If
End If
```

```

End If
End If
End If
End If
End If
End If

If KeyAscii = Asc(".") Then
    X = 2
End If
If KeyAscii = Asc(",") Then
    X = 3
End If
If KeyAscii = vbKeyReturn Then
    X = 4
End If

If X = 1 Then
    KeyAscii = 0
    MsgBox "Introduzca sólo números", vbInformation
End If
If X = 2 Then
    KeyAscii = Asc(".")
End If
If X = 3 Then
    KeyAscii = Asc(",")
End If
If X = 4 Then
    xi = xi + 1
    If xi = 14 Then
        Combo.SetFocus
        xi = 2
        Exit Sub
    End If
    Text1(xi).SetFocus
End If

```

3.1.2.2 Text1_Click

Cada vez que pasemos a un texto, mediante la variable xi sabemos en qué índice de texto estamos, para poder copia, pegar, cortar...

```
xi = Index
```

3.1.2.3 Text2_KeyPress

En el texto donde insertamos el código del tramo, cada vez que le demos al “Intro”, el programa sabe que debe de pulsar el botón cmdinsertar.

```
If KeyAscii = 13 Then
    cmdinsertar_Click
End If
```

3.1.3 Grids. Tablas

3.1.3.1 Grid1_KeyPress

Como en las cajas de texto, en las tablas sólo podemos introducir números y la “,”. Si pulsamos otra tecla aparece un mensaje.

```
If KeyAscii >= 47 And KeyAscii <= 57 Then
    Grid1.Text = Grid1.Text & Chr(KeyAscii)
ElseIf KeyAscii = 46 Then
    Grid1.Text = Grid1.Text & Chr(44)
ElseIf KeyAscii = 8 Then GoTo final
Else
    MsgBox "Introduzca sólo números", vbInformation
End If

final:
```

3.1.3.2 Grid1_KeyUp

Al soltar las teclas de Retroceso y Suprimir, borraremos una letra y borraremos todo el contenido de la casilla respectivamente.

```
Select Case KeyCode
    Case vbKeyDelete
        Grid1.Text = ""
    Case vbKeyBack
        If Len(Grid1.Text) > 0 Then
            Grid1.Text = Left(Grid1.Text, Len(Grid1.Text) - 1)
        End If
End Select
```

3.1.4 Botones

3.1.4.1 cmdbajar_Click

Utilizamos este botón para bajar un tramo de torre en la estructura de la torre.

Lo primero que se comprueba es que se ha elegido un tramo para desplazarlo. Si el tramo elegido es el último de todos no tiene sentido bajarlo más, de manera que sale del evento.

Lo que hacemos es guardar los datos del tramo siguiente elegido en la variable tramos(11), asignamos a la fila siguiente seleccionada los valores del tramo que se quiere bajar, y por último los valores de tramos(11) los pasamos a la fila seleccionada. Todo esto mediante un bucle For Next. Se selecciona toda la fila, para saber que tramo se ha bajado.

```
Dim tramos(11) As String
With Grid1
If Grid1.TextMatrix(.row, 2) = "" Then
    MsgBox "Seleccione un tramo de torre para desplazarlo.", vbInformation, "Aerogenerador"
    Exit Sub
End If
If Grid1.row = NumTorre Then Exit Sub
For j = 2 To 11
    tramos(j) = Grid1.TextMatrix(.row + 1, j)
    Grid1.TextMatrix(.row + 1, j) = Grid1.TextMatrix(.row, j)
    Grid1.TextMatrix(.row, j) = tramos(j)
Next
Grid1.row = Grid1.row + 1
Grid1.TopRow = Grid1.row - 1
Grid1.Col = 0
Grid1.ColSel = 11
final:
End With
```

3.1.4.2 cmdsubir_Click

Este evento, tiene como función subir un tramo de torre en la estructura. Es muy similar al anterior, cmdbajar. Se comprueba que se ha elegido un tramo y que este no sea el primero. Guardamos en la variable tramos(11) los datos del tramo anterior al elegido. En la fila del tramo anterior al elegido colocamos los datos del tramo elegido, y luego en la fila

seleccionada colocamos los datos de la variable tramos(11). Todo esto mediante un bucle For Next. Se selecciona la fila.

```
Dim tramos(11) As String
With Grid1
If Grid1.TextMatrix(.row, 2) = "" Then
    MsgBox "Seleccione un tramo de torre para desplazarlo.", vbInformation, "Aerogenerador"
    Exit Sub
End If
If Grid1.row = 1 Then Exit Sub
For j = 2 To 11
    tramos(j) = Grid1.TextMatrix(.row - 1, j)
    Grid1.TextMatrix(.row - 1, j) = Grid1.TextMatrix(.row, j)
    Grid1.TextMatrix(.row, j) = tramos(j)
Next
Grid1.row = Grid1.row - 1
If Grid1.row <> 1 Then
    Grid1.TopRow = Grid1.row - 1
End If
Grid1.Col = 0
Grid1.ColSel = 11
final:
End With
```

3.1.4.3 cmdeliminar_Click

Se utiliza este evento para eliminar un tramo de torre. Lo primero que se comprueba es que se haya elegido un tramo de torre. Tenemos dos casos. El primero es con la condición de que haya más de 10 filas, es decir que se hayan introducido más de un tramo. Y la segunda es que sólo haya 10 filas en la tabla. Según que caso se realizan unas operaciones o otras, pero básicamente es lo mismo, ya que el fin es eliminar toda la fila seleccionada. Por último con un bucle *For*, a la primera columna de la tabla, la que indica el número de tramos, le daremos el valor que corresponde.

```
If Grid1.Rows > 10 Then
With Grid1
    If Grid1.TextMatrix(.row, 2) = "" Then
        MsgBox "No hay ninguna fila seleccionada para eliminar.", vbExclamation, "Aerogenerador"
        Exit Sub
    Else
        If frmdatos.Grid1.row = 0 Then frmdatos.Grid1.row = 1
        .RemoveItem .row
    End If
End With
```

```

    NumTorre = NumTorre - 1
    frmdatos.Text1(1) = Val(frmdatos.Text1(1)) - 1

    If frmdatos.Grid1.row >= 2 Then
        frmdatos.Grid1.row = frmdatos.Grid1.row - 1
    End If
End If
End With
End If

GoTo fin

If Grid1.Rows = 10 Then
    With Grid1
        If Grid1.row > 1 Then
            If Grid1.TextMatrix(.row, 2) = "" Then
                MsgBox "No hay ninguna fila seleccionada para insertar.", vbExclamation, "Aerogenerador"
                Exit Sub
            End If
            If Grid1.TextMatrix(.row, 2) <> "" Then
                Grid1.RemoveItem .row
                Grid1.AddItem 10
                NumTorre = NumTorre - 1
                frmdatos.Text1(1) = Val(frmdatos.Text1(1)) - 1
            End If
        End If
    End With
End With
End If

fin:

For i = 1 To NumTorre
    frmdatos.Grid1.TextMatrix(i, 1) = i
Next

End Sub

```

3.1.4.4 cmdlimpiardatos_Click

Se utiliza para limpiar de datos las cajas de texto, de una manera rápida y sencilla. Utilizamos un bucle For Next que va desde la segunda caja de texto hasta la última.

```

For i = 2 To 13
    Text1(i).Text = ""
Next

```


3.1.4.5 cmdlimpiartramos_Click

Se utiliza para limpiar de datos de tramos la tabla, de una manera rápida y sencilla. El valor de la caja de texto “Nº de Tramos”, se queda a cero. Nos ayudamos de dos bucles For Next. El primero para movernos por las filas y el segundo para movernos por las columnas.

```
Grid1.row = 2
Do While Grid1.Rows > 10
    Grid1.RemoveItem Grid1.row
Loop
For i = 1 To 9
    For j = 1 To 12
        Grid1.TextMatrix(i, j) = ""
    Next
Next
NumTorre = 0
frmdatos.Text1(1) = ""
Grid1.row = 1
Grid1.TopRow = 1
```

3.1.4.6 cmdinsertar_Click

Se utiliza para introducir los tramos de torre en la tabla. Verificamos una serie de propiedades necesarias, para que no haya problemas al introducir el tramo, y así insertarlo en la fila del cuadro adecuada, y no haya ningún problema.

```
Dim row As Integer
row = frmdatos.Grid1.row
If Val(frmdatos.Text1(1)) = 0 Or frmdatos.Text1(1) = "" Then
    If row = 1 Then GoTo sigue
End If
If frmdatos.Grid1.TextMatrix(row + 1, 1) = "" Then
    frmdatos.Grid1.row = frmdatos.Grid1.row + 1
End If
sigue:
```

El código del tramo que se debe de introducir podemos conseguirlo de la lista desplegable o introduciendo el código en el Text2. Si se ha escrito algo en el Text2 toma lo escrito para insertar el tramo de torre, si no el valor de la lista desplegable. Llama a la función de insertar tramos de torre y a continuación deja en blanco el Text2 y elige el primer término de la lista desplegable.

```
ContadorTramos = 0
If frmdatos.Text2 <> "" Then
    CodigoTramos(1) = frmdatos.Text2
Else
    CodigoTramos(1) = frmdatos.Combo.Text
End If
CodigoTramos(ContadorTramos) = CodigoTramos(1)
Call InsertarDatosTorre
frmdatos.Text2 = ""
frmdatos.Combo.Text = frmdatos.Combo.List(0)
frmdatos.Text2.SetFocus
```

3.1.4.7 cmdcalculotorre_Click

Se verifica que se han introducido todos los datos, y que por lo menos hemos introducido un tramo de torre.

```
'Verificamos previamente que hayamos introducido todos los datos
For i = 2 To 13
    If Text1(i).Text = "" Then
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
        Text1(i).SetFocus
        Exit Sub
    End If
Next

'Verificamos que al menos se haya introducido un tramo de torre
If NumTorre = 0 Then
    MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"
    Exit Sub
End If
```

Introducimos en la variable Dat todos los datos y en la variable Tramo todas las propiedades de los tramos de torre. Ejecutamos la función Calculo. Cargamos el formulario frmcalculotorre y ocultamos el formulario principal frmdatos sin descargarlo.

```
'Introducimos en la Matriz Dat todos los datos de la grúa
For i = 1 To 13
    Dat(i) = Val(Text1(i).Text)
Next

'Introducimos en la variable Tramo todos los datos de torre
For i = 1 To Dat(i)
```

```

    For j = 1 To 11
        Tramo(i, j) = frmdatos.Grid1.TextMatrix(i + 1, j + 2)
    Next
Next

Call Calculo

frmcalculotorre.Show
frmdatos.Hide

```

3.1.4.8 cmdcalculobase_Click

Este evento es exactamente igual al anterior salvo que en este caso en vez de cargar el formulario frmcalculotorre se carga el formulario frmcalculobase.

```

'Verificamos previamente que hayamos introducido todos los datos
For i = 2 To 13
    If Text1(i).Text = "" Then
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
        Text1(i).SetFocus
        Exit Sub
    End If
Next
'Verificamos que al menos se haya introducido un tramo de torre
If NumTorre = 0 Then
    MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"
    Exit Sub
End If
'Introducimos en la Matriz Dat todos los datos de la grúa
For i = 1 To 13
    Dat(i) = Val(Text1(i).Text)
Next
'Introducimos en la variable Tramo todos los datos de torre
For i = 1 To Dat(i)
    For j = 1 To 11
        Tramo(i, j) = frmdatos.Grid1.TextMatrix(i + 1, j + 2)
    Next
Next

Call Calculo

frmcalculobase.Show
frmdatos.Hide

```

3.1.5 Botones Menú

3.1.5.1 mnuabrir_Click

Utilizamos este botón para abrir los archivos de texto o Excel que hayamos podido guardar antes. Con el control `CommonDialog1`, abrimos los archivos. Le asignamos una serie de propiedades al control. Comprobamos que no haya ningún error con el nombre del archivo, y guardamos su dirección en la variable `RutaFichero`. Se borran todos los datos del formulario con `cmdlimpiardatos_Click` y `cmdlimpiartramos_Click`. A continuación vemos si es un archivo de texto o de Excel, comprobando el final de la dirección del archivo, para llamar a la función adecuada, `AbrirExcel` o `AbrirTexto`. Con cualquier error se sale del evento.

```
Dim Texto As String
Dim posnombre As Integer
On Error GoTo error
With CommonDialog1
    .DialogTitle = "Abrir"
    .FileName = RutaFichero
    .Flags = cdlOFNFileMustExist
    .ShowOpen
    .CancelError = False

    If Err.Number = 0 Then
        If Len(.FileName) Then
            RutaFichero = .FileName
            posnombre = InStrRev(RutaFichero, ".")
            Texto = Mid(RutaFichero, posnombre + 1)
            cmdlimpiardatos_Click
            cmdlimpiartramos_Click

            'Comprobamos si es un archive Excel o de texto
            If Texto = "txt" Then
                Call AbrirTexto
            Else
                Call AbrirExcel
            End If
        End If
    End If
End With

error:
```

3.1.5.2 mnuabrirbase_Click

Utilizamos este botón para poder abrir una base de datos, que no sea por la de defecto. Con el control CommonDialog1, abrimos los archivos. Le asignamos una serie de propiedades al control. Comprobamos que no haya ningún error con el nombre del archivo, y guardamos su dirección en la variable RutaFichero. Cerramos la tabla y base de datos que este abierta. Con un pequeño mensaje nos informa que debemos de introducir el nombre de la tabla que deseamos abrir. El nombre de la tabla queda almacenado en la variable NombreTabla. Si no se ha producido ningún error va hasta el final del procedimiento, en cambio si al abrir la base ocurre un error, aparece un mensaje de advertencia del error, y nos dice que se abrirá de nuevo la base de datos por defecto. A las variables RutaFichero y NombreTabla damos el valor de la dirección de la base de datos y el nombre de la tabla que tiene al abrir la base de datos por defecto. Por último borramos todos los elementos del Combo1 del formulario y le asignamos todos los registros de la tabla que se haya abierto. Para conocer en qué dirección está la base de datos, junto al nombre del formulario escribimos esta dirección.

On Error GoTo error

With CommonDialog1

```
.DialogTitle = "Abrir Base"
.FileName = RutaFichero
.Flags = cdIOFNFileMustExist
.ShowOpen
.CancelError = False
```

If Err.Number = 0 Then

If Len(.FileName) Then

```
RutaBaseDatos = .FileName
RsTorre.Close
DbBaseDatos.Close
Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(RutaBaseDatos, True, True)
NombreTabla = InputBox("Introduzca el nombre de la tabla por favor", "Aerogenerador")
If NombreTabla = "" Then GoTo error
Set RsTorre = DbBaseDatos.OpenRecordset(NombreTabla, dbOpenDynaset)
```

End If

End If

End With

GoTo final

error:

```
MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical, "Aerogenerador"
Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal)
```

```
Set RsTorre = DbBaseDatos.OpenRecordset(TablaOriginal, dbOpenDynaset)
RutaBaseDatos = DirOriginal
NombreTabla = TablaOriginal
```

final:

```
Combo.Clear
RsTorre.MoveLast
RsTorre.MoveFirst
Combo.AddItem " "
```

```
For i = 1 To RsTorre.RecordCount
    Combo.AddItem RsTorre(0)
    RsTorre.MoveNext
Next
```

```
RsTorre.MoveFirst
```

```
frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos
```

3.1.5.3mn-guardar-texto_Click

Guardamos los datos y la configuración de la torre en un archivo de texto.

Verificamos que se hayan introducido todos los datos y además que por lo menos hayamos introducido un tramo de torre. Si no es así aparece un mensaje y se sale del evento.

Ejecutamos la función DatosTexto. Se carga el formulario frmhojaimprimir y se oculta el formulario principal frmdatos sin descargarlo.

'Verificamos que se hayan introducido todos los datos de la grúa

```
For i = 2 To 13
```

```
    If Text1(i).Text = "" Then
```

```
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
```

```
        Text1(i).SetFocus
```

```
        Exit Sub
```

```
    End If
```

```
Next
```

'Verificamos que se haya introducido por lo menos un tramo de torre

```
If NumTorre = 0 Then
```

```
    MsgBox "Introduzca al menos un tramo de torre", vbOKOnly, "Aerogenerador"
```

```
    Exit Sub
```

```
End If
```

```
Call DatosTexto  
frmhojaimprimir.Show  
frmdatos.Hide
```

3.1.5.4 mnuguardarexcel_Click

Es básicamente igual al evento mnuguardartexto_Click, solo que en este caso no se carga el formulario hojaimprimir ni se oculta el formulario frmdatos. En vez de ejecutar la función DatosTexto se ejecuta la función DatosExcel.

```
'Verificamos que se hayan introducido todos los datos de la grúa  
For i = 2 To 13  
    If Text1(i).Text = "" Then  
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"  
        Text1(i).SetFocus  
        Exit Sub  
    End If  
Next  
  
'Verificamos que se haya introducido por lo menos un tramo de torre  
If NumTorre = 0 Then  
    MsgBox "Introduzca al menos un tramo de torre", vbOKOnly, "Aerogenerador"  
    Exit Sub  
End If  
  
Call DatosExcel
```

3.1.5.5 mnuimprimir_Click

Ejecutamos mnuguardartexto_Click

```
mnuguardartexto_Click
```

3.1.5.6 mnusalir_Click

Salimos del programa.

```
End
```

3.1.5.7 mnuedicioncortar_Click

Típico evento para cortar algún dato de alguna caja de texto. Se limpia el portapapeles, se introduce en el portapapeles el texto seleccionado, y se borra el texto seleccionado de la caja de texto elegida.

```
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
Text1(xi).SelText = ""
```

3.1.5.8 mnuedicioncopiar_Click

Típico evento para copiar algún dato de alguna caja de texto. Se limpia el portapapeles, se introduce en el portapapeles el texto seleccionado, y se pone la propiedad Enabled del botón mnuedicionpegar a verdadero, para que pueda responder a eventos generados por el usuario.

```
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
mnuedicionpegar.Enabled = True
```

3.1.5.9 mnuedicionpegar_Clikc

Pegamos en el texto seleccionado los caracteres del portapapeles.

```
Text1(xi).SelText = Clipboard.GetText
```


3.1.6 Botones Barra de Herramientas

Al pulsar sobre algún botón de la barra de herramientas, es como si pulsásemos algún botón de la barra de menú. Excepto al pulsar el botón guardar. Al pulsarlo se verifica que hemos introducido todos los datos y por lo menos algún tramo de torre. A continuación carga el formulario frmguardar.

Select Case Button.Key

```

    Case "Open"
        mnuabrir_Click
    Case "Save"
        'Verificamos que se hayan introducido todos los datos de la grúa
        For i = 2 To 13
            If Text1(i).Text = "" Then
                MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
                Text1(i).SetFocus
            Exit Sub
        End If
    Next
    'Verificamos que al menos se haya introducido un tramo de torre
    If NumTorre = 0 Then
        MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"
    Exit Sub
End If
frmguardar.Show
Case "Print"
    mnuimprimir_Click
Case "Cut"
    mnuedicioncortar_Click
Case "Copy"
    mnuedicioncopiar_Click
Case "Paste"
    mnuedicionpegar_Click
'Case "Help"
' frmesquema.Show
End Select

```

3.2 FORMULARIO CÁLCULO DE LA TORRE. FRMCALCULOTORRE.

3.2.1 Formulario

3.2.1.1 Form_Load

Con este evento cargamos el formulario principal. En el describimos las propiedades de los dos Grids (tablas). Sólo en modo de ejecución se puede describir ciertas propiedades de estas tablas. La anchura de columnas, textos que pueden aparecer en cada casilla, orientación de texto...

De manera que lo primero que se hace al cargar el formulario es describir las tablas. Utilizamos la función With, para hacer referencia a la tabla en cuestión, y luego simplemente aplicamos la propiedad elegida. Describimos el texto que debe aparecer en la casilla deseada y a continuación la anchura de las columnas. Con un bucle For Next, aplicamos a todas las columnas una alineación de texto central.

For i = 1 To 2

With Grid1(i)

```
.TextArray(1) = "Tramo"
.TextArray(2) = "m"
.TextArray(3) = "Kg"
.TextArray(4) = "Cortante Kg"
.TextArray(5) = "Kg·m"
.TextArray(6) = "cm"
.TextArray(7) = "Montante"
.TextArray(8) = "Celosía Sup"
.TextArray(9) = "Celosía Inf"
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1150
.ColWidth(3) = 1150
.ColWidth(4) = 1300
.ColWidth(5) = 1400
.ColWidth(6) = 1400
.ColWidth(7) = 1400
.ColWidth(8) = 1400
.ColWidth(9) = 1400
```

End With

With Grid2(i)

```
.TextArray(1) = "Nº"
.TextArray(2) = "Altura"
.TextArray(3) = "Peso"
.TextArray(4) = "Esfuerzo"
.TextArray(5) = "Momento"
```

```
.TextArray(6) = "Flecha"
.TextArray(7) = "Tensión - Kg/cm²"
.TextArray(8) = "Tensión - Kg/cm²"
.TextArray(9) = "Tensión - Kg/cm²"
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1150
.ColWidth(3) = 1150
.ColWidth(4) = 1300
.ColWidth(5) = 1400
.ColWidth(6) = 1400
.ColWidth(7) = 1400
.ColWidth(8) = 1400
.ColWidth(9) = 1650
.MergeRow(0) = True

End With
Next

For i = 1 To 9
    Grid1(1).ColAlignment(i) = 4
    Grid2(1).ColAlignment(i) = 4
    Grid1(2).ColAlignment(i) = 4
    Grid2(2).ColAlignment(i) = 4
Next
```

3.2.2 Botones

3.2.2.1 cmdexcel_Click

Al pulsar este botón se guardan todos los resultados obtenidos, tanto los referidos al cálculo de la torre como los de la base. Se avisa de que el proceso puede llevar su tiempo mediante un mensaje. Si se da a cancelar sale del evento. Cambiamos el icono del ratón para que el usuario sepa que el ordenador está trabajando, y se ejecuta la función `TodoResultadoExcel`. Por último volvemos a darle al icono del ratón su valor normal.

```
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then
    MousePointer = 11
    Call TodoResultadosExcel
    MousePointer = 0
End If
```

3.2.2.2 cmdtexto_Click

Igual que el evento cmdexcel_Click se guardan todos los resultados, pero esta vez en archivo de texto. Se avisa de que el proceso puede llevar su tiempo, con la posibilidad de cancelarlo. Se cambia el icono del ratón, para reflejar que el ordenador está trabajando. Ejecutamos las funciones DatosTexto, ResultadosTorreTexto y ResultadosBaseTexto. Por último volvemos a darle al icono del ratón su valor normal. Se carga el formulario frmhojaimprimir

```
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then
    MousePointer = 11
    Call DatosTexto
    Call ResultadosTorreTexto
    Call ResultadosBaseTexto
    MousePointer = 0
    frmhojaimprimir.Show
End If
```

3.2.3 Botones de Menú

3.2.3.1 mnuguardartexto_Click

Guardamos los datos, la configuración de la torre y resultados del cálculo de torre en un archivo de texto. Cambiamos el icono del ratón. Ejecutamos la función DatosTexto y ResultadosTorreTexto. Se carga el formulario frmhojaimprimir. Por último volvemos a cambiar el icono del ratón.

```
MousePointer = 11
Call DatosTexto
Call ResultadosTorreTexto
MousePointer = 0
frmhojaimprimir.Show
```

3.2.3.2 mnuguardarexcel_Click

Igual que mnuguardartexto_Click guardamos los datos, la configuración de la torre y resultados del cálculo de torre en un archivo de Excel. Todo esto ejecutando la función ResultadosTorreExcel.

```
Call ResultadosTorreExcel
```

3.2.3.3 mnuimprimir_Click

Ejecutamos el evento mnuguardartexto_Click

`mnuguardar_Click`

3.2.3.4 mnusalir_Click

Salimos del programa

`End`

3.2.3.5 mnuedicioncortar_Click

Típico evento para cortar algún dato de alguna caja de texto. Se limpia el portapapeles, se introduce en el portapapeles el texto seleccionado, y se borra el texto seleccionado de la caja de texto elegida.

`Clipboard.Clear`

`Clipboard.SetText Text1(xi).SelText`

`Text1(xi).SelText = ""`

3.2.3.6 mnuedicioncopiar_Click

Típico evento para copiar algún dato de alguna caja de texto. Se limpia el portapapeles, se introduce en el portapapeles el texto seleccionado, y se pone la propiedad Enabled del botón mnuedicionpegar a verdadero, para que pueda responder a eventos generados por el usuario.

`Clipboard.Clear`

`Clipboard.SetText Text1(xi).SelText`

`mnuedicionpegar.Enabled = True`

3.2.4 Botones Barra de Herramientas

Según que botón elijamos ejecuta su correspondiente botón de la barra de menú.

```
Select Case Button.Key
    Case "Cut"
        mnuedicioncortar_Click
    Case "Copy"
        mnuedicioncopiar_Click
    Case "Paste"
        'mnuedicionpegar_Click
    Case "Print"
        mnuimprimir_Click
    Case "Save"
        frmguardar.Show
End Select
```

3.3 FORMULARIO CÁLCULOS DE LA BASE. FRMCALCULOBASE

3.3.1 Botones de Menú

3.3.1.1 mnuguardartexto_Click

Guardamos los datos, la configuración de la torre y resultados del cálculo de base en un archivo de texto. Cambiamos el icono del ratón. Ejecutamos la función DatosTexto y ResultadosBaseTexto. Se carga el formulario frmhojaimprimir. Por último volvemos a cambiar el icono del ratón.

```
ousePointer = 11
Call DatosTexto
Call ResultadosBaseTexto
MousePointer = 0
frmhojaimprimir.Show
```

3.3.1.2 mnuguardarexcel_Click

Igual que mnuguardartexto_Click guardamos los datos, la configuración de la torre y resultados del cálculo de torre en un archivo de Excel. Todo esto ejecutando la función ResultadosBaseExcel.

[Call ResultadosBaseExcel](#)

3.3.1.3 mnuimprimir_Click

Ejecutamos el evento mnuguardartexto_Click

[Mnuguardar_Click](#)

3.3.1.4 mnusalir_Click

Salimos del programa

[End](#)

3.3.1.5 mnuedicioncortar_Click

Típico evento para cortar algún dato de alguna caja de texto. Se limpia el portapapeles, se introduce en el portapapeles el texto seleccionado, y se borra el texto seleccionado de la caja de texto elegida.

[Clipboard.Clear](#)

[Clipboard.SetText Text1\(xi\).SelText](#)

[Text1\(xi\).SelText = ""](#)

3.3.1.6 mnuedicioncopiar_Click

Típico evento para copiar algún dato de alguna caja de texto. Se limpia el portapapeles, se introduce en el portapapeles el texto seleccionado, y se pone la propiedad Enabled del botón mnuedicionpegar a verdadero, para que pueda responder a eventos generados por el usuario.

```
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
mnuedicionpegar.Enabled = True
```

3.3.1.7 mnuedicionpegar_Click

Pegamos en el texto seleccionado los caracteres del portapapeles.

```
Text1(xi).SelText = Clipboard.GetText
```

3.3.2 Botones Barra de Herramientas

Según que botón elijamos ejecuta su correspondiente botón de la barra de menú.

```
Select Case Button.Key  
    Case "Cut"  
        mnuedicioncortar_Click  
    Case "Copy"  
        mnuedicioncopiar_Click  
    Case "Paste"  
        'mnuedicionpegar_Click  
    Case "Print"  
        mnuimprimir_Click  
    Case "Save"  
        frmguardar.Show  
End Select
```


3.4 FORMULARIO HOJA DE IMPRESIÓN. FRMHOJAIMPRIMIR

3.4.1 Botones de Menú

3.4.1.1 mnuguardartexto_Click

Guardamos los datos que aparecen en el texto. Utilizamos el control `CommonDialog1` para ello. Le aplicamos una serie de propiedades a este control. A continuación creamos el archivo de texto con la función `Open`. Con cualquier error se sale del evento.

```

Dim RutaFichero As String
Dim nFic As Long
On Error GoTo final

    With CommonDialog1
        .DialogTitle = "Guardar configuración"
        .FileName = RutaFichero
        .Flags = cdlOFNOverwritePrompt
        .Filter = "Documentos de texto (*.txt)|*.txt|Todos los archivos|*.*"
        .ShowSave
        .CancelError = False
        If Len(.FileName) Then
            RutaFichero = .FileName
            nFic = FreeFile
            Open RutaFichero For Output As nFic
                Print #nFic, Text1.Text
            Close nFic
        End If
    End With

final:

```

3.4.1.2 mnuimprimir_Click:

3.4.1.3mnusalir_Click

Se sale del programa

End

3.4.1.4 mnuedicioncortar_Click

Típico evento para cortar algún dato de alguna caja de texto. Se limpia el portapapeles, se introduce en el portapapeles el texto seleccionado, y se borra el texto seleccionado de la caja de texto elegida.

```
Clipboard.Clear  
Clipboard.SetText Text1.SelText  
Text1.SelText = ""
```

3.4.1.5 mnuedicioncopiar_Click

Típico evento para copiar algún dato de alguna caja de texto. Se limpia el portapapeles, se introduce en el portapapeles el texto seleccionado, y se pone la propiedad Enabled del botón mnuedicionpegar a verdadero, para que pueda responder a eventos generados por el usuario.

```
Clipboard.Clear  
Clipboard.SetText Text1.SelText  
mnuedicionpegar.Enabled = True
```

3.4.1.6 mnuedicionpegar_Click

Pegamos en el texto seleccionado los caracteres del portapapeles.

```
Text1.SelText = Clipboard.GetText
```

3.4.1.7 mnuedicionselecciontodo_Click

Se selecciona todo el texto del cuadro de texto Text1

```
Text1.SetFocus  
Text1.SelStart = 0  
Text1.SelLength = Len(Text1.Text)
```

3.4.2 Botones Barra de Herramientas

Según que botón elijamos ejecuta su correspondiente botón de la barra de menú.

```
Select Case Button.Key
    Case "Save"
        mnguardar_Click
    Case "Cut"
        mnuedicioncortar_Click
    Case "Copy"
        mnuedicioncopiar_Click
    Case "Paste"
        mnuedicionpegar_Click
    Case "Print"
        'mnuimprimir_Click
End Select
```

3.5 FORMULARIO GUARDAR COMO. FRMGUARDAR

Este formulario tiene 3 botones. Excel, Texto y Cancelar.

3.5.1 Excel

Según que formulario este cargado y visible, se ejecutarán unos procedimientos para grabar los resultados.

```
If frmdatos.Visible = True Then  
Call DatosExcel  
End If
```

```
If frmcalculotorre.Visible = True Then  
Call ResultadosTorreExcel  
End If
```

```
If frmcalculobase.Visible = True Then  
Call ResultadosBaseExcel  
End If
```

3.5.2 Texto

Según que formulario este cargado y visible, se ejecutarán unos procedimientos para grabar los resultados. Se carga el formulario frmhojaimprimir.

```
If frmdatos.Visible = True Then  
Call DatosTexto  
End If
```

```
If frmcalculotorre.Visible = True Then  
Call ResultadosTorreTexto  
End If
```

```
If frmcalculobase.Visible = True Then  
Call ResultadosBaseTexto  
End If
```

```
frmhojaimprimir.Show
```

4 MÓDULOS

Hasta ahora sólo hemos explicado todo lo relativo a los formularios. Pero como se ha comentado con la pequeña idea general sobre el programa Visual Basic, los programas constan de formularios y módulos. En estos módulos hemos escrito varias funciones.

Nuestro programa tiene 3 módulos:

- Cálculos: En el hemos escrito todas las funciones y algoritmos referidos al programa de cálculo en sí. Todo ello se ha explicado con detenimiento en sus respectivas secciones.
- Funciones: En el hemos escrito 4 procedimientos distintos. AbrirTexto, AbrirExcel, AbrirBase, InsertarDatosTorre. Estos se explican a continuación.
- GuardarDatos: A la hora de grabar los datos de entrada y los resultados tenemos varias opciones de hacerlo. Mediante estos procedimientos lo conseguimos.

Explicamos cada uno de los siguientes procedimientos de una manera rápida y sencilla, para que el usuario tenga una pequeña idea de que es lo que hace el programa.

Como hemos dicho en el módulo Cálculos se han escrito todo lo relativo al cálculo de la torre y cálculo de la base más el pequeño estudio de cimentación. Todo ello se ha explicado antes.

4.1 MÓDULO FUNCIONES

Dentro del módulo Funciones tenemos una declaración de variables que se hacen públicas para que se puedan utilizar en todo el programa.

- RutaFichero: Se utiliza para guardar la dirección del archivo de texto o Excel que se ha abierto.
- RutaBaseDatos: Se utiliza para guardar la dirección de la base de datos que se ha abierto.
- DirOriginal: Se utiliza para guardar la dirección de la base de datos que se abre por defecto, por si ocurre un error para poder seguir ligado a una base de datos de torre.
- NombreTabla: Guardamos en esta variable el nombre de la tabla de la base de datos que se esté utilizando
- TablaOriginal: Se utiliza para guardar el nombre de la tabla que se ha abierto por defecto, por si ocurre un error poder abrir la base de datos de torre.
- NumTorre: Guardamos en esta variable el número de tramos de torre que se ha introducido.

- DbBaseDatos: Variable con la que nos referimos a la base de datos.
- RsTorre: Variable con la que nos referimos a la tabla de la base de datos.
- nFic: Variable que se utiliza al abrir un archivo de texto.
- lineatexto: Variable que se utiliza al abrir un archivo de texto, en el que se lee cada línea del archivo abierto.
- CodigoTramos(80): Se utiliza para guardar los códigos de tramo de torre.
- ContadorTramos: Se utiliza para llevar la cuenta del número de tramo que se ha insertado.

Luego se hace una declaración de variables de tipo integer para poder utilizarlas en todo el módulo.

```
Public RutaFichero As String
Public RutaBaseDatos As String
Public DirOriginal As String
Public NumTorre As Integer
Public NombreTabla As String
Public TablaOriginal As String
Public DbBaseDatos As Database
Public RsTorre As Recordset
Public nFic As Long
Public lineatexto As String
Public CodigoTramos(80) As String
Public ContadorTramos As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer
```

4.1.1 AbrirExcel

Utilizamos este procedimiento para abrir cualquier archivo Excel que se haya guardado con anterioridad, para así poder recuperar esos datos y realizar pruebas de cálculo.

Lo primero que se hace es declarar una serie de variables. La más importante de ellas es la de apli, con ella diremos al programa que queremos que sea de tipo Excel. Utilizamos las variable direcciónbase, tabla y posbase para obtener la dirección de la base de datos así como el nombre de la tabla, que vamos a abrir

```
Dim apli As Excel.Application
Dim direccionbase As String
Dim tabla As String
Dim posbase As Integer
```

Creamos el objeto Excel mediante la función Set, abrimos el archivo de Excel con la dirección que se ha obtenido antes y hemos guardado en la variable RutaFichero. Seleccionamos la primera hoja y la celda “b3”.

```
Creo el objeto
Set apli = New Excel.Application
'Abro el archivo seleccionado la hoja y celda deseada
apli.Workbooks.Open RutaFichero
apli.Sheets(1).Select
apli.Range("b3").Select
```

Al elegir la celda “b3”, el programa se sitúa en ella, y a continuación pasamos los datos de torre y datos de cimentación al formulario principal. Para ello el programa se mueve de fila en fila en el archivo de Excel. Para que no haya problemas de cálculo cambiamos las comas, por puntos de decimal en los datos de las cajas de texto.

```
For i = 1 To 10
    frmdatos.Text1(i).Text = apli.ActiveCell.Offset(i, 0).Value
Next

apli.Range("b17").Select

For i = 1 To 3
    frmdatos.Text1(i + 10).Text = apli.ActiveCell.Offset(i, 0).Value
Next

For i = 1 To 13
    frmdatos.Text1(i) = Replace(frmdatos.Text1(i), ",", ".")
Next
```

Seleccionamos la celda “e5”, y guardamos el código de cada tramo de torre en la variableCodigoTramos.

```
apli.Range("e5").Select

For i = 1 To Val(frmdatos.Text1(1))
    CodigoTramos(i) = apli.ActiveCell.Offset(i, 0)
ext
```

Antes de insertar los datos de torre, debemos de abrir la base de datos con la que se hizo la configuración de torre. Para ello mediante una serie de instrucciones y con distintas

funciones obtenemos tanto la dirección de la base de datos como el nombre de la tabla que aparece en el archivo de Excel y se llama a la función AbrirBase. Con cualquier error se abriría la base de datos por defecto.

```
direccionbase = apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, -1)
posbase = InStrRev(direccionbase, "Dirección: ")
direccionbase = Mid(direccionbase, posbase + 11)
RutaBaseDatos = direccionbase
direccionbase = apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 4, -1)
osbase = InStrRev(direccionbase, "Tabla: ")
tabla = Mid(direccionbase, posbase + 7)
NombreTabla = tabla
```

Call AbrirBase

Con el código de los tramos guardado y la base de datos abierta, ejecutamos la función InsertarDatosTorre.

Para que se ejecute bien el procedimiento, se ponen una serie de condiciones, como por ejemplo que la fila en que se inserte el tramo sea igual al número del tramo...

'Insertamos los tramos de pluma llamando al procedimiento InsertarDatosTorre

```
For k = 1 To Val(frmdatos.Text1(1))
  ContadorTramos = k
  If k > frmdatos.Grid1.Rows - 2 Then
    MsgBox ("Se ha producido un error al insertar los datos de torre"), vbCritical, "Aerogenerador"
    GoTo final:
  End If
  CodigoTramos(ContadorTramos) = CodigoTramos(k)
  frmdatos.Grid1.row = k
  Call InsertarDatosTorre
Next
```

Como cada vez que se ejecuta el procedimiento InsertarDatosTorre, en la caja de texto del formulario principal “Número de Tramos” se suma uno, al final tenemos el doble con lo que dividimos por dos ese valor. Borraremos todos los valores que puedan aparecer por debajo del último tramo de torre de la tabla que tenemos en el formulario principal. Haciendo esto conseguimos que todos los datos se hayan introducido correctamente. Por último se cierra el libro y el archivo de Excel. Los datos para el cálculo están listos con lo que el botón cmdcalculotorre obtiene el foco.


```

frmdatos.Text1(1) = Val(frmdatos.Text1(1)) / 2
    For i = Val(frmdatos.Text1(1)) + 1 To frmdatos.Grid1.Rows - 1
        frmdatos.Grid1.TextMatrix(i, 1) = ""
    Next
final:

apli.Workbooks.Close
apli.DisplayAlerts = False
apli.Quit

frmdatos.cmdcalculotorre.SetFocus

RutaBaseDatos = direccionbase
  
```

4.1.2 AbrirTexto

Utilizamos este procedimiento para abrir cualquier archivo de texto que se haya guardado con anterioridad, para así poder recuperar esos datos y realizar pruebas de cálculo.

Declaramos una serie de variables que se utilizan a lo largo del código.

- lineasdatos(1000): Guardamos cada línea de texto en esta variable.
- lineastramos(1000): Guardamos cada línea de tramo de torre en esta variable.
- nlineas: Guardamos en esta variable el número de líneas que tenemos.
- Pos(1000): Guardamos la posición de un carácter determinado “=” en cada línea de texto que aparezca. Con este valor sabemos en qué línea tenemos un valor de dato de torre y de dato de cimentación.
- PosBase(500): Guardamos la posición de un carácter determinado “Dirección” en cada línea de texto que aparezca. Con este valor sabemos en qué línea tenemos la dirección de la base de datos así como el nombre de la tabla, que será la siguiente línea.
- PosTramos: Guardamos la posición de la palabra “Código”. Sabemos así donde en qué línea empieza los datos de los tramos de torre.
- cont: Variable que utilizamos para
- base: Pondremos como condición que si este valor es igual a False, se siga buscando valores y sigamos dentro de un bucle específico. En teoría la dirección de la base de datos está escrita después de todos los códigos de tramos de torre, con lo que se saldrá del bucle en cuanto hayamos encontrado la dirección de la base de datos. Conseguimos así que el procedimiento sea más rápido.

- direcciónbase: Guardamos en esta variable la dirección de la base de datos que aparece en el archivo de texto.
- tabla: Guardamos en esta variable el nombre de la tabla de la base de datos que vayamos a abrir.

```
Dim lineasdatos(1000) As String
Dim lineastorre(1000) As String
Dim nlineas As Integer
Dim Pos(1000) As Integer
Dim PosBase(500) As Integer
Dim PosTramos As Integer
Dim cont As Integer
Dim base As Boolean
Dim direccionbase As String
Dim tabla As String
```

Ponemos a cero las variables para que no haya errores. Y damos a la variable base el valor de False.

Abrimos el archivo de texto mediante la función Open y mientras no se llegue al final del archivo, por cada línea de texto sumamos la unidad a la variable nlineas. Cerramos el archivo.

```
nFic = FreeFile
nlineas = 0
PosTramos = 0
base=False

Open RutaFichero For Input As nFic
  While Not EOF(1)
    Line Input #nFic, lineatexto
    nlineas = nlineas + 1
  Wend
Close #nFic
```

Abrimos de nuevo el archivo. Mediante un bucle guardamos en la variable lineasdatos(i) cada línea de texto que se lea. Lo primero que se comprueba es si la variable base es igual a True, de ser así se sale del bucle, ya que habremos guardado todos los datos necesarios. Buscamos la palabra “Tabla”, para saber en qué líneas se encuentra la dirección de la base de datos y el nombre de la tabla, guardamos este valor en la variable PosBase(i). Cuando el valor de PosBase(i) sea distinto de cero, quiere decir que estamos en la línea deseada donde podemos encontrar el nombre de la tabla. De manera que con la función

Mid, obtenemos el nombre de la tabla de la base de datos y la guardamos en la variable tabla. En una línea anterior a esta, está la dirección de la base de datos. Sabiendo esto obtenemos la dirección de la base de datos que la guardamos en la variable direccionbase. Como se ha encontrado la dirección y la tabla, damos a la variable base = True, para que no se vuelva a repetir estas instrucciones.

A continuación se busca en cada línea de texto el carácter "=", para saber en qué línea se encuentran los datos de entrada y en qué posición se encuentra dentro de la línea, guardamos este valor en la variable Pos(i). Se busca también la palabra "Código", para conocer en qué línea empiezan los datos de tramo de torre, la guardamos en la variable PosTramos. Se utilizan para detectar en qué posición de la cadena están los caracteres "=" y "Código" las funciones InStrRev y InStr respectivamente. Una vez encontrado el valor de la línea en que aparece "Código" le damos al valor de PosTramos el valor de la línea es decir:

PosTramos = i

i el valor de PosTramos es distinto de 0, quiere decir que la línea se ha encontrado, con lo que, no hace falta volver a realizar la búsqueda de los caracteres "=" y "Código", con lo que se sigue el bucle, para poder pasar cada línea de texto a la variable lineasdatos(i).

Si se supera el valor de 500 líneas, se sale del bucle, ya que quiere decir que estamos leyendo ya los datos debidos a los resultados.

Open RutaFichero For Input As nFic

'Repite el bucle hasta el final del archivo.

While Not EOF(1)

For i = 1 To nlineas

If i > 500 Then GoTo salimosbucle

Lee el carácter en la variable.

Line Input #nFic, lineatexto

lineasdatos(i) = lineatexto

If base = True Then GoTo salimosbucle

PosBase(i) = InStrRev(lineasdatos(i), "Tabla: ")

If PosBase(i) <> 0 Then

tabla = Mid(lineasdatos(i), PosBase(i) + 7)

PosBase(i - 1) = InStrRev(lineasdatos(i - 1), "Dirección: ")

direccionbase = Mid(lineasdatos(i - 1), PosBase(i - 1) + 11)

base = True

End If

```

      If PosTramos <> 0 Then GoTo sigue
      Pos(i) = InStrRev(lineasdatos(i), "=")
      PosTramos = InStr(lineasdatos(i), "Código")
      If PosTramos <> 0 Then
        PosTramos = i
      End If
sigue:
      Next
Wend

salimosbucle:

```

Ponemos a cero la variable cont. Con un nuevo bucle desde la primera hasta la última línea, encontramos que líneas tienen datos y las colocamos en orden. Es decir si la línea “Número de tramos” es la 20ª la pasamos a la 1ª, para que concuerde con su número de dato. Para ello nos ayudamos de la variable cont. Para saber si es una línea de dato, con las posiciones que antes hemos encontrado de Pos(i), imponemos la condición de que los últimos caracteres sean números. Si no es un número a la variable cont le sumamos la unidad. Cerramos el archivo.

```

cont = 0
For i = 1 To nlineas
  If Not IsNumeric(Mid(lineasdatos(i), Pos(i) + 1, 7)) Then
    cont = cont + 1
  End If

  If IsNumeric(Mid(lineasdatos(i), Pos(i) + 1, 7)) Then
    lineasdatos(i - cont) = Mid(lineasdatos(i), Pos(i) + 1, 7)
  End If
Next

Close #nFic

```

Pasamos el valor de los datos, que ahora están guardados en la variable lineasdatos(i), a las cajas de texto del formulario principal. Para que no haya problemas de cálculo quitamos las comas de decimales y los pasamos a puntos.

```

For i = 1 To 13
  lineasdatos(i) = Replace(lineasdatos(i), ",", ".")
  frmdatos.Text1(i) = Replace(Val(lineasdatos(i)), ",", ".")
Next

```

Tenemos el número de tramos de torre que posee la torre. Con un bucle desde el primer hasta el último tramo, como sabemos en qué línea empiezan los datos de tramo de torre, simplemente pasamos el valor de esa línea a la variable `lineastramos(i)`. Una vez tenemos esto, nos quedamos sólo con los 10 primeros caracteres de las líneas, utilizando la función `Left`. A continuación le quitamos todos los espacios en blanco que puedan tener esos 10 caracteres, tanto por la derecha como por la izquierda. Utilizamos para ello la función `Trim`. Lo que queda será el código del tramo.

```
For i = 1 To Val(frmdatos.Text1(1))
    lineastorre(i) = lineasdatos(PosTramos + i + 2)
    lineastorre(i) = Left(lineastorre(i), 10)
    lineastorre(i) = Trim(lineastorre(i))
Next
```

Antes de insertar los datos de torre, debemos de abrir la base de datos. Hemos ardao la dirección de esta en la variable `direccionbase` y el nombre de la tabla en la variable `tabla`.

Damos a las variables `RutaBaseDatos` y `NombreTabla` estos valores, y llamamos al procedimiento `AbrirBase`. Con cualquier error se abrirá la base de datos por defecto.

```
RutaBaseDatos = direccionbase
NombreTabla = tabla

Call AbrirBase
```

Una vez que tenemos los códigos de los tramos de torre, ejecutamos un bucle desde el primer hasta el último tramo de torre y llamamos a la función `InserDatosTorre`.

```
'Insertamos los tramos de pluma llamando al procedimiento InsertarDatosTorre
For k = 1 To Val(frmdatos.Text1(1))
    ContadorTramos = k
    CodigoTramos(ContadorTramos) = lineastorre(k)
    frmdatos.Grid1.row = k
    Call InsertarDatosTorre
Next
```

Como pasa al abrir un archivo de Excel, cada vez que se ejecuta el procedimiento

```
frmdatos.Text1(1) = Val(frmdatos.Text1(1)) / 2
For i = Val(frmdatos.Text1(1)) + 1 To frmdatos.Grid1.Rows - 1
```

```
frmdatos.Grid1.TextMatrix(i, 1) = ""
```

Next

```
frmdatos.cmdcalculotorre.SetFocus
```

InsertarDatosTorre, en la caja de texto del formulario principal “Número de ramos” se suma uno, al final tenemos el doble con lo que dividimos por dos ese valor. Borramos todos los valores que puedan aparecer por debajo del último tramo de torre de la tabla que tenemos en el formulario principal. Haciendo esto conseguimos que todos los datos se hayan introducido correctamente. Por último los datos para el cálculo están listos con lo que el botón cmdcalculotorre obtiene el foco.

4.1.3 InsertarDatosTorre

Esta función se utiliza para insertar los datos de torre en la tabla del formulario principal.

Se declaran una serie de variables todas del tipo string. Las utilizamos para buscar el tramo de torre que queremos insertar dentro de la tabla de la base de datos.

Dim comilla As String

Dim criterio As String

Dim codigo As String

Dim marca As String

Damos a las variables comilla y marca sus valores. A la variable comilla le damos el valor de Chr(34), que equivale a las comillas “. A la variable marca le asignamos una de las propiedades de la tabla, Bookmark. La cual nos indica cual fue la última posición de fila que se registro buscando los códigos de los tramos de torre.

```
comilla = Chr(34)
```

```
marca = RsTorre.Bookmark
```

Para que no haya problemas de introducción de datos en la tabla y no ocurra ningún error, por si esta elegida la primera fila de la tabla del formulario principal, en la cual no se pueden introducir datos, se salta todo el procedimiento y va hasta el final de él. Lo mismo ocurre si no se ha introducido ningún código de tramo en la caja de texto del formulario principal. Un mensaje de error aparece y se salta todo el procedimiento llevándonos al

final.

If frmdatos.Grid1.row = 0 Then GoTo final

If CodigoTramos(ContadorTramos) = "" Then

MsgBox "Introduzca el Código del tramo de torre", vbExclamation, "Aerogenerador"

frmdatos.Grid1.row = frmdatos.Grid1.row - 1

GoTo final

End If

Se podrán introducir como mucho 80 tramos de torre, por ello se pone como condición para poder insertar estos que la cantidad de NumTorre sea menor de 80.

A la variable código le asignamos el código del tramo de torre que queremos insertar. A la variable criterio le asignamos pues el criterio de búsqueda. Esta última es especial pero aplicando una serie de propiedades y características propias de Visual Basic al trabajar con bases de datos de Access, resulta una manera sencilla y rápida de encontrar el código deseado.

If NumTorre < 80 Then

codigo = CodigoTramos(ContadorTramos)

criterio = "Código like" & comilla & codigo & comilla

Movemos el registro de búsqueda a la primera fila de la tabla de base de datos. Con el criterio de búsqueda que hemos escrito, se empieza el registro del código del tramo. Si no se encuentra un mensaje de error aparece avisándonos de ello y se va hasta el final del procedimiento. En cambio si se encuentra insertaremos los datos en la tabla del formulario principal. Una vez más para que no ocurra ningún error establecemos que si la fila seleccionada es la 0, se selecciona la 1º y se inserta una nueva fila en la tabla del formulario principal. Con un bucle desde la primera hasta la última columna, vamos pasando los valores de la tabla de la base de datos a la tabla de formulario principal. A cada primera columna de la tabla le damos el valor de la posición del tramo de torre. Sumamos a la variable NumTorre y al valor de la caja de texto del formulario “Número de Tramos” la unidad. Por estética del formulario establecemos que si se han introducido más de 8 tramos de torre, la tabla del formulario tendrá ese número de filas más 4.

RsTorre.MoveFirst

RsTorre.FindFirst criterio

If RsTorre.NoMatch Then

MsgBox "No se encontró el tipo de tramo especificado. Vuelva a intentarlo.", vbInformation, "Aerogenerador"

RsTorre.Bookmark = marca

frmdatos.Grid1.row = frmdatos.Grid1.row - 1

```

    GoTo final
Else
    Dim row As Integer
    row = frmdatos.Grid1.row
    If row = 0 Then row = 1
    frmdatos.Grid1.AddItem "", row
        For j = 1 To 11
            frmdatos.Grid1.TextMatrix(row, j + 1) = RsTorre(j - 1)
        Next
    frmdatos.Grid1.TextMatrix(row, 1) = row
    NumTorre = NumTorre + 1
    frmdatos.Text1(1) = Val(frmdatos.Text1(1)) + 1
    If NumTorre > 8 Then
        frmdatos.Grid1.Rows = NumTorre + 4
    End If

End If

Else

    Si el valor de tramos de torre es mayor que 80 aparece un mensaje de error
    avisándonos de ello. Se borra lo escrito dentro de la caja de texto. Por último nos movemos
    al primer registro de la tabla de la base de datos, y damos a la primera columna de la tabla
    del formulario el valor de la posición del tramo de torre.

    MsgBox "No se pueden introducir más tramos de torre.", vbCritical, "Aerogenerador"
    frmdatos.Text2.Text = ""
    End If

final:

RsTorre.MoveFirst
For i = 1 To Val(frmdatos.Text1(1))
    frmdatos.Grid1.TextMatrix(i, 1) = i
Next

```


4.1.4 AbrirBase

Utilizamos esta función para abrir la base datos del archivo de texto o de Excel que estemos abriendo.

Con cualquier error se aparece un mensaje de advertencia, diciendo que se abrirá la base de datos por defecto.

Lo primero que se hace es cerrar tanto la base de datos como la tabla. Con las direcciones guardadas en las variables RutaBaseDatos y NombreTabla abrimos la base de datos y su respectiva tabla. Si no se produce ningún error salta el mensaje de advertencia. Limpia el Combo1 del formulario de datos y pasa todos los códigos de tramos de torre a este. Por último junto al nombre del formulario de datos, escribimos la dirección de la base de datos para saber cuál es la que estamos utilizando.

```

On Error GoTo error
RsTorre.Close
DbBaseDatos.Close

Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(RutaBaseDatos)
Set RsTorre = DbBaseDatos.OpenRecordset(NombreTabla, dbOpenDynaset)
GoTo final:

error:
MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical, "Aerogenerador"
Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal)
Set RsTorre = DbBaseDatos.OpenRecordset(TablaOriginal, dbOpenDynaset)
RutaBaseDatos = DirOriginal
NombreTabla = TablaOriginal

final:

frmdatos.Combo.Clear
RsTorre.MoveLast
RsTorre.MoveFirst
frmdatos.Combo.AddItem " "
For i = 1 To RsTorre.RecordCount
    frmdatos.Combo.AddItem RsTorre(0)
    RsTorre.MoveNext
Next

RsTorre.MoveFirst

frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos

```

4.2 MÓDULO GUARDARDATOS

Por último tenemos el módulo GuardarDatos. En el tenemos los procedimientos que utilizamos para guardar los datos. Podemos diferenciar dos grandes grupos, los referidos a los archivos de texto y los de Excel. En los primeros lo que hacemos es escribir los datos en el texto del formulario frmhojaimprimir. Con los archivos Excel en cambio lo que se hace es crear un archivo de Excel y escribir datos y resultados en él.

Archivos de texto:

- DatosTexto: Escribimos en el caja de texto del formulario frmhojaimprimir los datos. Igual que en el formulario tenemos pos un lado los datos de torre, datos de cimentación y por último datos de los tramos de torre.
- ResultadosBaseTexto: Escribimos en el caja de texto del formulario frmhojaimprimir los resultados que hemos obtenido de la base y del pequeño estudio de cimentación.
- ResultadosTorreTexto: Escribimos en el caja de texto del formulario frmhojaimprimir los resultados que hemos obtenido del cálculo de la torre.
- TodoResultadosTexto: Escribimos en el caja de texto del formulario frmhojaimprimir todos los resultados y datos de entrada.

Cada vez que demos a guardar resultados en los formularios frmcalculotorre y frmcalculobase, cuando se carga el formulario frmhojaimprimir, aparecen tanto los resultados como los datos de entrada.

Archivos de Excel:

- DatosExcel: Abrimos un archivo Excel y escribimos los datos. Igual que en el formulario tenemos pos un lado los datos de torre, datos de cimentación y por último datos de los tramos de torre.
- ResultadosBaseExcel: Abrimos un archivo Excel y escribimos los resultados que hemos obtenido de la base y del pequeño estudio de cimentación.

- ResultadosTorreExcel: Abrimos un archivo Excel y escribimos los resultados que hemos obtenido del cálculo de la torre.
- TodoResultadosExcel: Abrimos un archivo Excel y escribimos todos los resultados y datos de entrada.

Al grabar los resultados en Excel tendremos en distintas hojas los datos de entrada y los resultados. En la primera de ellas tenemos los datos y en las siguientes los resultados. Esto se hace para que cuando queramos abrir este tipo de archivos el programa sepa que tiene que abrir siempre la primera de ellas.

CAPÍTULO 3:

Manual Usuario

1 FORMULARIO PRINCIPAL.

En este formulario introducimos los datos de cálculo.

- Podemos introducir todos los datos desde el teclado. Para desplazarnos de una casilla de texto a otra podemos pulsar tabulador o Intro.

Cuando introduzcamos los datos de torre, podemos introducir tanto el código del tramo escribiéndolo en la casilla que esta junto al botón insertar o eligiendo el tramo del desplegable. Después de haber elegido una de las dos opciones debemos pulsar “Insertar”. El valor de la casilla “Número de Tramos”, aumentará en 1.

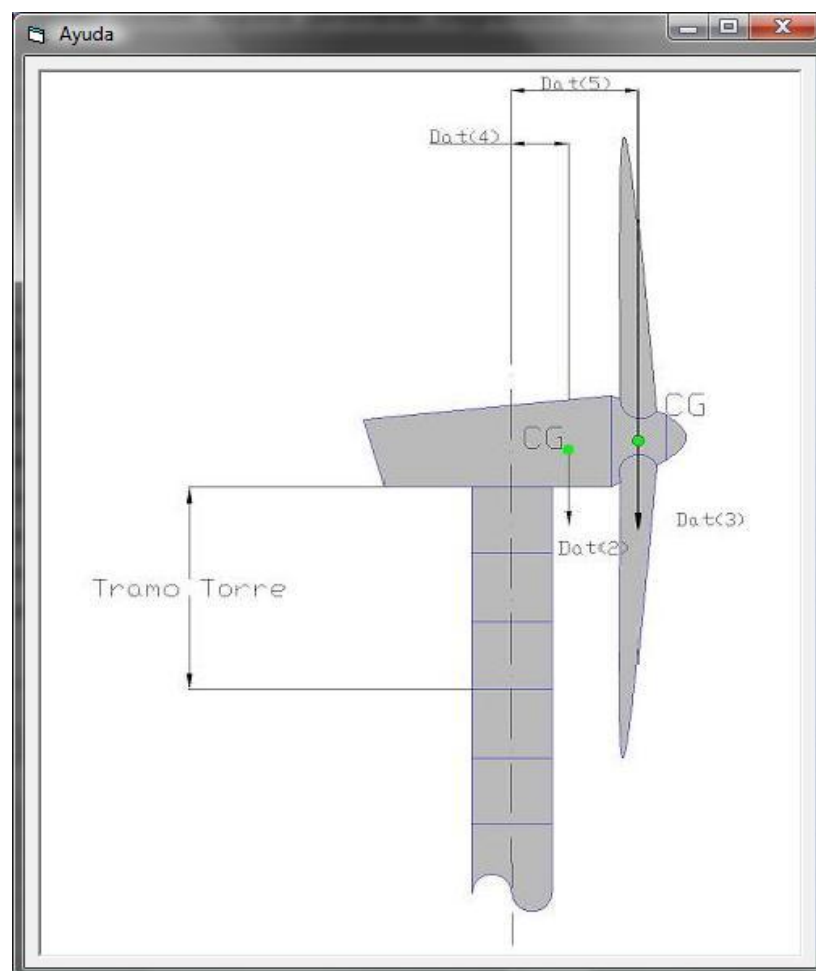
Los botones “Subir” y “Bajar”, sirven para desplazar los tramos de torre. Si queremos eliminar algún tramo presionaremos el botón “Eliminar”, y si queremos que desaparezcan todos “Limpiar Tramos”. Para desplazar o eliminar un tramo primero debemos de elegir en la tabla el tramo deseado.

Nota: ¡¡Los tramos de torre se introducen por orden empezando desde arriba!!

-Abriendo un archivo existente. Barra de Menú. Archivo. Abrir. Tenemos dos tipos de archivos que puede leer el programa. Archivos Excel o Texto.

- Arrastrando un archivo tanto Excel o de Texto sobre el formulario.

Puede ocurrir que algún usuario no tenga muy claro, que se quiere decir con una serie de datos de introducción como por ejemplo: con distancia centro de gravedad góndola, distancia hasta aspas... Para eso si pulsamos “Ayuda” en el Barra de Menú o pinchando en el símbolo de interrogación, aparecerá un pequeño esquema del aerogenerador, haciendo una pequeña idea de que son ese tipo de datos.



Una vez introducidos todos los datos de cálculo, podremos guardarlos en un archivo de Excel o Texto e incluso imprimirlos. Barra de Herramientas, Archivo, Guardar Como...

Si pulsamos el típico símbolo de guardar en la Barra de Herramientas, aparece un pequeño cuadro tal como se indica en la figura. En el tenemos las opciones de guardar los datos en texto o Excel, o simplemente pulsando cancelar saldremos del procedimiento.



Si algún dato no se ha introducido o por lo menos no hemos insertado un tramo de torre, un mensaje de advertencia aparece, con lo que no podemos realizar el procedimiento de guardar o imprimir.

Una de las opciones que tenemos en la Barra de Menú, en el desplegable Archivo, es la de Abrir Base de Datos.

Cada usuario puede tener una base de datos propia, en la que haya creado una serie de tramos propios, siempre con las propiedades geométricas establecidas. De esta manera puede realizar diferentes pruebas y ver cómo reacciona el aerogenerador con distintas composiciones de torre. La opción de Abrir Base de Datos nos da la opción de realizar estas pruebas. Como cualquier archivo que se quiera abrir, se elige la base de datos deseada. A continuación debemos de introducir el nombre de la tabla que queramos abrir.

Nota: ¡¡Las Bases de Datos deben de ser creadas con Microsoft Access 2003 o anterior!!

Nota: ¡¡El nombre del código de cada tramo de torre no debe de exceder de 9 letras!!

Hemos introducido todos los datos de cálculo. Tenemos dos apartados de cálculo. Por un lado tenemos los cálculos de la torre, y por otro los cálculos de base junto con el pequeño estudio de cimentación. Simplemente pulsando el botón correspondiente obtendremos el formulario deseado. Si algún dato no se ha introducido o por lo menos no hemos insertado un tramo de torre, un mensaje de advertencia aparece, con lo que no podemos realizar los cálculos.

[illegible]

- Barra de Menú. Archivo, Guardar Como...

- En Archivo Excel. Obtenemos los resultados que aparecen en el formulario y los datos que se han introducido.

92

Aparece el cuadro antes mencionado donde nos da la posibilidad de guardar los resultados en texto o en Excel.

Pulsando los botones del formulario “Todo Excel” o “Todo Texto”, obtendremos todos los resultados, tanto los referidos al cálculo de la torre como los de la base más el pequeño estudio de cimentación. Aparecerá un pequeño mensaje en el que advertirá de que el proceso puede costarle unos segundos, con la opción de cancelar el procedimiento.

3 FORMULARIO CÁLCULO DE BASE

Aerogenerador - Cálculo Base

Archivo Edición

CÁLCULOS BASE

REACCIONES

	1	2	3	4
CASO 1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
CASO 2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

CASO 1

Par de Vuelco (N·m)

Reacción Horizontal (N)

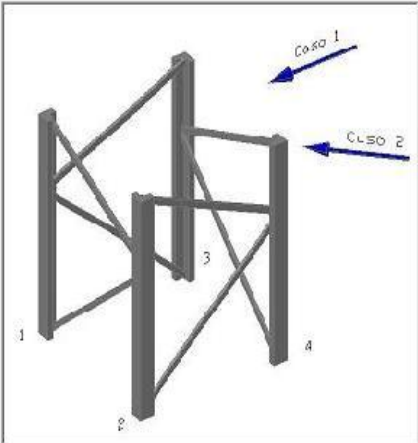
Reacción Máxima (N)

CASO 2

Par de Vuelco (N·m)

Reacción Horizontal (N)

Reacción Máxima (N)



CÁLCULOS CIMENTACIÓN

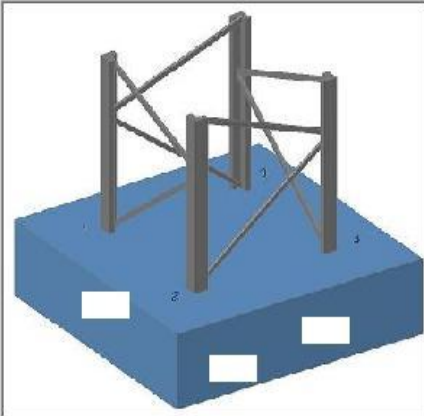
Longitud Zapata (m)

Altura Zapata (m)

Tensión Zapata (Kg/m²)

Tipo Distribución de Tensiones

Todo Excel Todo Texto



Hay varias opciones para guardar los resultados.

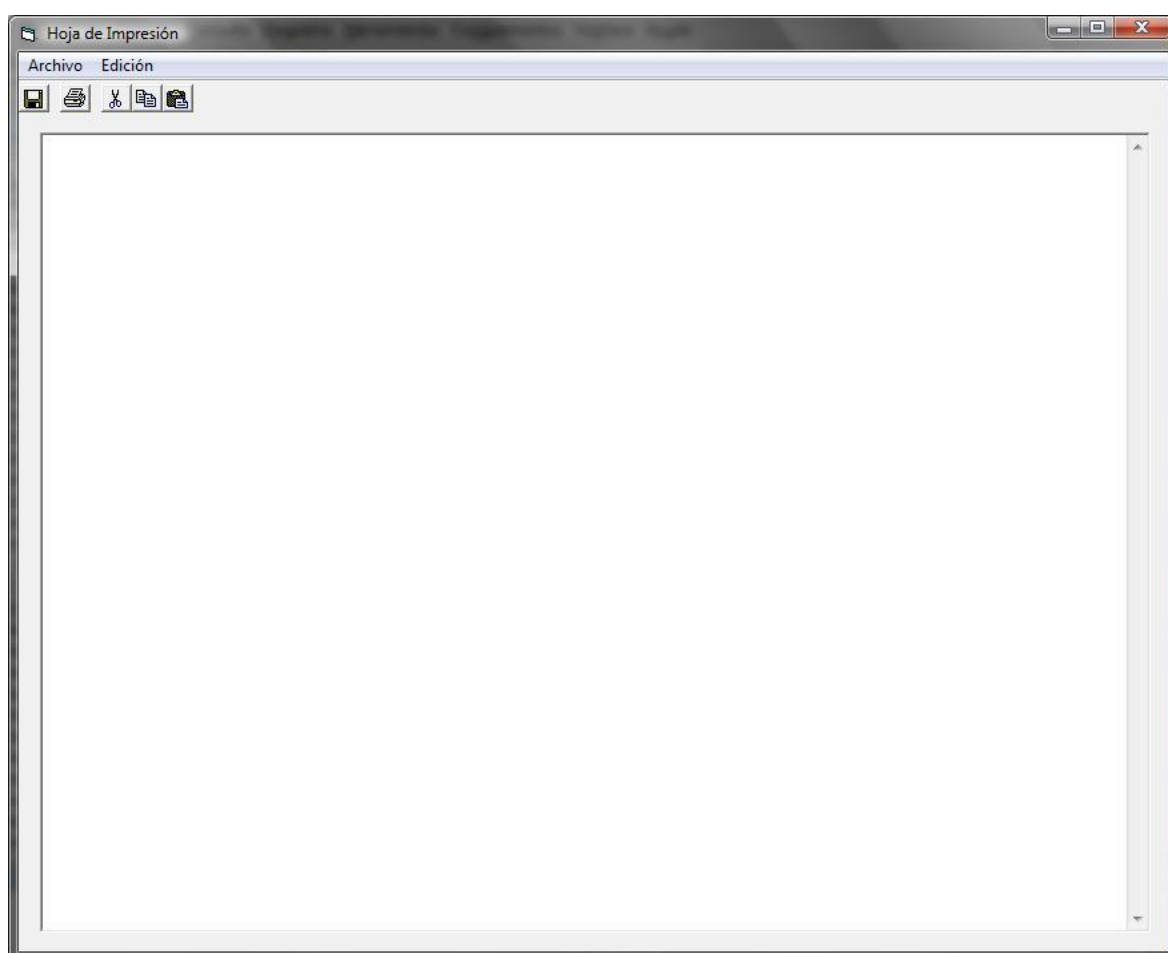
- Barra de Menú. Archivo, Guardar Como...
- En Archivo de Texto. Obtenemos los resultados que aparecen en el formulario y los datos que se han introducido.
- En Archivo Excel. Obtenemos los resultados que aparecen en el formulario y los datos que se han introducido.

- Barra de Herramientas. Guardar.

Aparece el cuadro antes mencionado donde nos da la posibilidad de guardar los resultados en texto o en Excel.

Pulsando los botones del formulario “Todo Excel” o “Todo Texto”, obtendremos todos los resultados, tanto los referidos al cálculo de la torre como los de la base más el pequeño estudio de cimentación. Aparecerá un pequeño mensaje en el que advertirá de que el proceso puede costarle unos segundos, con la opción de cancelar el procedimiento.

4 FORMULARIO HOJA DE IMPRESIÓN



A la hora de guardar los datos en texto, este formulario se carga. Mediante este formulario podemos guardar los datos o imprimirlos. Todos los botones son típicos de cualquier “Bloc de notas”

Nota: ¡¡Si deseamos guardar alguna nota o escribir algo en él no hay problema, pero siempre después de los datos introducidos, eso incluye los tramos de torre!!



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

***“Cálculo de aerogenerador en celosía mediante
software basado en Visual Basic”***

ANEXO

Javier Vergara Lecue

Tutores: Cesar Díaz de Cerio

Índice

Página

PARTE 1: INTRODUCCIÓN Y OBJETIVOS..... 15

1.1 INTRODUCCIÓN..... 16

1.2 OBJETIVO DEL PROYECTO 19

PARTE 2: CONCEPTOS TEÓRICOS 21

CAPÍTULO 1: Esquema de los Cálculos 22

1.1 INTRODUCCIÓN..... 23

1.2 ANTECEDENTES DE LOS RRHH EN ESPAÑA 24

1.2.1 ETAPA ADMINISTRATIVA 24

1.2.2 ETAPA DE LAS RELACIONES HUMANAS 24

1.2.3 ETAPA DE LAS RELACIONES LABORALES 25

1.2.4 ETAPA DE LOS RECURSOS HUMANOS 26

1.2.5 ETAPA DE LA ORGANIZACIÓN ESTRATÉGICA 26

1.3 ESTRUCTURACIÓN Y FUNCIONES ACTUALES..... 28

CAPÍTULO 2: Prácticas relacionadas con la Dirección 31

98

<i>2.1 DISEÑO DE LA ORGANIZACIÓN</i>	<i>32</i>
2.1.1 ESTRUCTURA BUROCRÁTICA.....	32

En este documento están recogidos todos los algoritmos y listado de procedimientos del programa, como la normas del EUROCÓDIGO UTILIZADAS.

1 LISTADO DE LOS PROCESMIENTOS

1.1 MÓDULO CÁLCULOS

1.1.1 Declaración de variables

```

Option Explicit
Public Const Pi = 3.141592654
Public Tramo(80, 1 To 12) As Variant
Public Dat(1 To 30) As Double
Dim NInc(1 To 2) As Single
Dim AltInc(100) As Double
Dim AltViento(100) As Double
Dim MomentosCaso1(100, 4) As Double
Dim MomentosCaso2(100, 4) As Double
Dim MomentosSecunCaso1(100, 4) As Double
Dim MomentosSecunCaso2(100, 4) As Double
Dim Cr(100) As Double 'Coeficiente de Rugosidad
Dim Ct(100) As Double 'Coeficiente Topográfico
Dim Iv(100) As Double 'Intensidad de Turbulencia
Dim Ce(100) As Double 'Coeficiente de Exposición
Dim Cd As Double 'Coeficiente Dinámicos
Dim CF(80) As Double 'Coeficiente de Fuerza/Empuje
Dim CfMedio As Double 'Coeficiente de Fuerza medio
Dim g As Double 'Factor de pico
Dim Vm(100) As Double 'Velocidad Media Incremento
Dim Pre(100) As Double 'Presión del viento en cada incremento
Dim FVientoCaso1(100) As Double 'Fuerza del viento en cada incremento
Dim FVientoCaso2(100) As Double
Dim FVientoElementos(10) As Double 'Fuerza del viento en elementos, buje, caja, aspas...
Dim Peso(0 To 100) As Double 'Peso de cada incremento, teniendo en cuenta los pesos que tiene por encima
Dim EsfCortanteVientoCaso1(0 To 100) As Double 'Esfuerzo cortante en cada incremento
Dim EsfCortanteVientoCaso2(0 To 100) As Double 'Esfuerzo cortante en cada incremento
Dim Kt As Single 'Factor de terreno
Dim Zo As Single 'Longitud de rugosidad
Dim Zmin As Single 'Altura mínima
Dim Ancho As Single 'Anchura base
Dim Lh As Single 'Espesor Zapata
Dim e As Single 'Factor de categoría del terreno
Dim FlechasCaso1(100, 4) As Double
  
```

Dim FlechasCaso2(100, 4) As Double
 Dim EsfCortante(2) As Double
 Dim MomentoPeso As Double
 Dim MomentoTotal(2) As Double
 Dim MomentoSecundarioTotal(2) As Double
 Dim Resultados(30) As Variant 'Guardaremos los resultados en esta variable
 Dim Variables(30) As Double 'Guardaremos distintos valores momentos, esfuerzos cortantes...

1.1.2 Función Calculo.

Sub Calculo()
 'ENV 1991-2-4 Acciones del Viento Tabla 8.1 Categoría del Terreno 1
 Kt = 0.17
 Zo = 0.01
 e = 0.13
 Zmin = 2
 'Anchura Base
 Ancho = Tramo(Dat(1), 4)

 Call Incremento
 Call CoeficientesViento
 Call FuerzaViento
 Call CalculoMomentos_Esfuerzos
 Call CalculoFlechas
 Call CalculoMomentosSecundarios
 Call CalculoEstructura

 End Sub
 Call CalculoBase

1.1.3 Función cálculo de incrementos.

Sub Incremento()
 Dim i As Integer
 Dim j As Integer
 Dim k As Integer

 'Con estas dos variables, sabemos cuántos incrementos tenemos en cada Tramo
 NInc(1) = Int(100 / (Dat(1) + 2)) 'Nº de Incrementos por cada tramo de torre
 NInc(2) = Int(Dat(1) + 2) * NInc(1) 'Nº de Incrementos Totales

 k = 1

 'División de la Torre
 For i = 1 To Dat(1)
 For j = 1 To NInc(1)

Javier Vergara

$\text{AltInc}(k) = \text{Tramo}(i, 3) / \text{NInc}(1)$

$k = k + 1$

Next

Next

End Sub

1.1.4 Función cálculo de coeficientes de viento

1.1.5 Función cálculo de la fuerza del viento en cada incremento

```

Sub FuerzaViento()
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim h As Double 'Variable que se utiliza para calcular la altura respecto del suelo de cada incremento

h = 0

'Calcularemos la altura total de la torre
For i = 1 To Dat(1)
    h = h + Tramo(i, 3)
Next

'Calcularemos, el coeficiente de rugosidad, topográfico y de exposición de cada incremento.
'Así como la velocidad y presión media. Por último la fuerza que ejerce el viento en la torre para cada incremento

k = 1
For i = 1 To Dat(1)
    For j = 1 To NInc(1)
        AltViento(k) = h - AltInc(k) / 2
        'Coeficiente de rugosidad
        If AltViento(k) > Zo Then
            Cr(k) = Kt * Log(AltViento(k) / Zo)
        Else
            Cr(k) = Kt * Log(Zmin / Zo)
        End If
        'Coeficiente topográfico
        'Debo de preguntar
        Ct(k) = 1
        'Intensidad de turbulencia
        Iv(k) = Kt / (Cr(k) * Ct(k))
        'Coeficiente de exposición
        Ce(k) = Cr(k) ^ 2 * Ct(k) ^ 2 * (1 + 2 * g * Iv(k))
        'Velocidad media incremento' Faltarían por introducir
        Vm(k) = Dat(9) * Cr(k) * Ct(k)
        'Presión en cada incremento
        Pre(k) = 1.25 * 0.5 * Vm(k) ^ 2
        'Fuerza del viento en cada incremento
        FVientoCaso1(k) = Pre(k) * Ce(k) * Cd * CF(i) * Tramo(i, 9) / NInc(1)

        h = h - AltInc(k)
        k = k + 1
    Next j
Next i

```

Next

Next

End Sub

1.1.6 Función cálculo de momentos y esfuerzos en cada incremento.

Sub CalculoMomentos_Esfuerzos()

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim l As Integer

Dim Dist(100) As Double

Dim DistBucle As Double

'Distancia respecto de la base de la parte superior del punto de incidencia del viento en buje/caja y aspas

Dist(0) = Dat(7) / 2

'Momento Causado por el peso del buje y las aspas ya que están "descentrados"

MomentoPeso = Dat(2) * Dat(4) + Dat(3) * Dat(5)

For k = 1 To NInc(2)

Dist(k) = Dist(k - 1) + AltInc(k)

DistBucle = AltInc(k) / 2

'CASO 1

'Momento causado por viento en buje/caja

MomentosCaso1(k, 1) = FVientoElementos(1) * Dist(k)

'Momento causado por viento en aspas

MomentosCaso1(k, 2) = FVientoElementos(2) * Dist(k)

'Momento causado por el momento del peso causado por buje/caja/aspas

MomentosCaso1(k, 3) = MomentoPeso

'Momento causado por viento en torre

MomentosCaso1(k, 4) = FVientoCaso1(k) * DistBucle

'CASO 2

'Momento causado por viento en buje/caja

MomentosCaso2(k, 1) = FVientoElementos(1) * Dist(k)

'Momento causado por viento en aspas

MomentosCaso2(k, 1) = FVientoElementos(2) * Dist(k)

'Momento causado por el momento del peso causado por buje/caja/aspas

MomentosCaso2(k, 3) = MomentoPeso

'Momento causado por viento en torre

MomentosCaso2(k, 4) = FVientoCaso2(k) * DistBucle

'Con este bucle sumaremos a cada "incremento", el momento causado por el viento que esté por encima de cada "incremento"

```

For l = k + 1 To NInc(2)
    DistBucle = DistBucle + AltInc(l)
    MomentosCaso1(l, 4) = MomentosCaso1(l, 4) + FVientoCaso1(k) * DistBucle
    MomentosCaso2(l, 4) = MomentosCaso2(l, 4) + FVientoCaso2(k) * DistBucle
Next

'Cálculo del esfuerzo cortante de cada incremento
EsfCortanteVientoCaso1(k) = EsfCortanteVientoCaso1(k - 1) + FVientoCaso1(k)
EsfCortanteVientoCaso2(k) = EsfCortanteVientoCaso2(k - 1) + FVientoCaso2(k)

Next

k = 0

For i = 1 To Dat(1)
    For j = 1 To NInc(1)
        k = k + 1
        'Calculo del peso de cada incremento
        Peso(k) = Peso(k - 1) + Tramo(i, 5) / NInc(1)
    Next
Next

End Sub

```

1.1.7 Función cálculo de flechas.

```

Sub CalculoFlechas()
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
    Dim l As Integer
    Dim Factor(100) As Double 'En esta variable guardaremos el valor de EIz de cada incremento, con el cambio de unidades
    Dim DistEje As Double
    Dim AlturaInc As Double
    Dim Dist As Double

    'Cálculo de las flechas producidas por el viento sobre la torre.
    'Aplicaremos el segundo teorema de Moore para hallar las flechas.
    'Flecha A-B=Integral(Mz/EIz)x dx

    'Cálcularemos el momento de inercia de cada incremento.

    k = 1

    For i = 1 To Dat(1)
        AlturaInc = 0
        For j = 1 To NInc(1)

```

```

Factor(k) = Tramo(i, 6) * Tramo(i, 4) ^ 2 * 21000
k = k + 1
Next
Next
k = 0

'Aplicación de Mohr
For j = 1 To 4
  For i = 1 To NInc(2) 'Desde el primer hasta el último "incremento"
    Dist = 0
    For k = i To NInc(2)
      Dist = Dist + (AltInc(k) / 2)
      'Momento Esfuerzos*Dist*Alt/F
      FlechasCaso1(i, j) = FlechasCaso1(i, j) + MomentosCaso1(k, j) * Dist * AltInc(k) / Factor(k)

      FlechasCaso2(i, j) = FlechasCaso2(i, j) + MomentosCaso2(k, j) * Dist * AltInc(k) / Factor(k)
      Dist = Dist + (AltInc(k) / 2)
    Next
  Next
Next
End Sub

```

1.1.8 Función cálculo de momentos secundarios

```

Sub CalculoMomentosSecundarios()
  Dim i As Integer
  Dim j As Integer
  Dim k As Integer

  'Cálculo de los momentos secundarios

  'Una vez que hemos calculado las flechas, calcularemos los momentos secundarios que hay en cada incremento
  'por la excentricidad de los pesos, derivada por las deformaciones

  For k = 1 To 4 '(Nº de "Momentos" distintos que tenemos en la base, por cada una de las flechas de la anterior lista)
    For i = 1 To NInc(2)
      MomentosSecunCaso1(i, k) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso1(1, k) - FlechasCaso1(i, k))
      MomentosSecunCaso2(i, k) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso2(1, k) - FlechasCaso2(i, k))

      'Con este bucle sumaremos a cada "incremento", el momento causado por el viento que esté por encima de cada "incremento"
      For j = 1 To i
        MomentosSecunCaso1(i, k) = MomentosSecunCaso1(i, k) + Peso(j) * Dat(10) * (FlechasCaso1(j, k) - FlechasCaso1(i, k))
        MomentosSecunCaso2(i, k) = MomentosSecunCaso2(i, k) + Peso(j) * Dat(10) * (FlechasCaso2(j, k) - FlechasCaso2(i, k))
      Next
    Next
  Next
End Sub

```



```

Next
Next

MomentoSecundarioTotal(1) = MomentosSecunCaso1(NInc(2), 1) + MomentosSecunCaso1(NInc(2), 2) +
MomentosSecunCaso1(NInc(2), 3) + MomentosSecunCaso1(NInc(2), 4)

MomentoSecundarioTotal(2) = MomentosSecunCaso2(NInc(2), 1) + MomentosSecunCaso2(NInc(2), 2) +
MomentosSecunCaso2(NInc(2), 3) + MomentosSecunCaso2(NInc(2), 4)

End Sub

```

1.1.9 Función cálculo de estructura

```

Sub CalculoEstructura()
Dim i As Integer
Dim ii As Integer
Dim j As Integer
Dim ki As Integer
Dim l As Integer
Dim EsfUnitMont(6) As Double
Dim EsfUnitDS(4) As Double
Dim EsfUnitDI(4) As Double
Dim MOM2Caso1(4) As Double
Dim MOM2Caso2(4) As Double
Dim T(9, 4) As Double
Dim aux(3) As Single

frmcalculotorre.Grid1(1).Rows = Dat(1) + 2
frmcalculotorre.Grid1(2).Rows = Dat(1) + 2

For i = 1 To Dat(1)
    j = i * NInc(1)

    'Puesta a cero de las Variables
    For l = 1 To 6
        EsfUnitMont(l) = 0
        MOM2Caso1(l) = 0
        MOM2Caso2(l) = 0
    Next
    For l = 1 To 4
        EsfUnitDI(l) = 0
        EsfUnitDS(l) = 0
    Next
    For l = 1 To 9
        T(l, 1) = 0
        T(l, 2) = 0
    Next

```

Call Esfuerzo_Unitario(i, EsfUnitMont, EsfUnitDS, EsfUnitDI)
 Call MomentosSecundarios_Unitario(j, MOM2Caso1, MOM2Caso2)

ki = i * NInc(1)

'CÁLCULO CASO 1-DIRECCIÓN HORIZONTAL

aux(1) = MOM2Caso1(1) + MOM2Caso1(2) + MOM2Caso1(3) + MOM2Caso1(4) 'Momentos Secundarios
 aux(1) = MomentosSecunCaso1(ki, 1) + MomentosSecunCaso1(ki, 2) + MomentosSecunCaso1(ki, 3) + MomentosSecunCaso1(ki, 4)
 T(1, 1) = i
 T(2, 1) = 0 'Altura
 T(3, 1) = (Dat(2) + Dat(3) + Peso(ki)) * Dat(10) 'Peso - Para los dos casos el mismo
 T(4, 1) = FVientoElementos(1) + FVientoElementos(2) + EsfCortanteVientoCaso1(ki) 'Fuerza Cortante
 T(5, 1) = MomentosCaso1(ki, 1) + MomentosCaso1(ki, 3) + MomentosCaso1(ki, 5) + MomentosCaso1(ki, 7) + aux(1)
 T(6, 1) = FlechasCaso1(ki, 1) + FlechasCaso1(ki, 2) + FlechasCaso1(ki, 3) + FlechasCaso1(ki, 4)
 T(7, 1) = T(5, 1) * EsfUnitMont(1) + T(3, 1) * EsfUnitMont(3) 'Tensión Montante
 T(8, 1) = T(4, 1) * EsfUnitDS(1) 'Tensión Dia Sup
 T(9, 1) = T(4, 1) * EsfUnitDI(1) 'Tensión Dia Inf

'CÁLCULO CASO 2-DIRECCIÓN DIAGONAL

aux(2) = MOM2Caso2(1) + MOM2Caso2(2) + MOM2Caso2(3) + MOM2Caso2(4) 'Momentos Secundarios
 aux(2) = MomentosSecunCaso2(ki, 1) + MomentosSecunCaso2(ki, 2) + MomentosSecunCaso2(ki, 3) + MomentosSecunCaso2(ki, 4)
 T(1, 2) = i
 T(2, 2) = 0 'Altura
 T(3, 2) = (Dat(2) + Dat(3) + Peso(ki)) * Dat(10) 'Peso - Para los dos casos el mismo
 T(4, 2) = FVientoElementos(1) + FVientoElementos(2) + EsfCortanteVientoCaso1(ki) 'Fuerza Cortante
 T(5, 2) = MomentosCaso2(ki, 1) + MomentosCaso2(ki, 3) + MomentosCaso2(ki, 5) + MomentosCaso2(ki, 7) + aux(2)
 T(6, 2) = FlechasCaso2(ki, 1) + FlechasCaso2(ki, 2) + FlechasCaso2(ki, 3) + FlechasCaso2(ki, 4)
 T(7, 2) = T(5, 2) * EsfUnitMont(2) + T(3, 2) * EsfUnitMont(3) 'Tensión Montante
 T(8, 2) = T(4, 2) * EsfUnitDS(2) 'Tensión Dia Sup
 T(9, 2) = T(4, 2) * EsfUnitDI(2) 'Tensión Dia Inf

1.1.10 Función cálculo de esfuerzos unitarios.

Sub Esfuerzo_Unitario(i, EsfUnitMont, EsfUnitDS, EsfUnitDI)

Dim Alfa As Double

'Ángulos de las Diagonales

Alfa = Atn(2 * Tramo(i, 3) / (Tramo(i, 2) * Tramo(i, 4)))

'Esfuerzos en el montante por flexión pura en dirección transversal a la sección. Caso 1

EsfUnitMont(1) = 1 / (2 * Tramo(i, 4) * Tramo(i, 6))

'Esfuerzos en la diagonal superior e inferior por esfuerzos cortante en dirección transversal a la sección. Caso 1

EsfUnitDS(1) = 1 / (2 * Tramo(i, 7))

EsfUnitDI(1) = 1 / (4 * Cos(Alfa) * Tramo(i, 8))

'Esfuerzos en el montante por flexión pura en dirección diagonal a la sección. Caso 2

EsfUnitMont(2) = 1 / (Sqr(2) * Tramo(i, 4) * Tramo(i, 6))

'Esfuerzos en la diagonal superior e inferior por esfuerzos cortante en dirección diagonal a la sección. Caso 2

EsfUnitDS(2) = 1 / (2 * Sqr(2) * Tramo(i, 7))

EsfUnitDI(2) = 1 / (4 * Sqr(2) * Cos(Alfa) * Tramo(i, 8))

'Esfuerzos en el montante debido al peso propio de la estructura, teniendo en cuenta del peso de buje/caja y aspás.

EsfUnitMont(3) = 1 / (4 * Tramo(i, 6))

End Sub

1.1.11 Función cálculo de momentos secundarios unitarios

Sub MomentosSecundarios_Unitario(j, MOM2Caso1, MOM2Caso2)

Dim l As Integer

Dim X As Integer

'MOMENTOS SECUNDARIOS

'Momentos Producidos por la Excentricidad del Peso Debida a las Distintas Flechas

'Al deformarse la torre, los pesos de esta quedan descentrados y producen unos momentos

'que aunque de escasa importancia también producen un esfuerzo

'Calcularemos el momento producido por el peso de la parte superior con excentricidad igual a la diferencia de flechas

'entre el primer incremento, que es el más alto, y el incremento más bajo del tramo

'y a continuación el momento que produce el peso de cada intervalo, con la diferencia de flechas de cada incremento

'respecto a la base

'Los Momentos Secundarios FS solo son útiles en los casos MOMFS(1), MOMFS(3) y MOMFS(6)

For l = 1 To 4

'If j < NinC(2) Then

'MOM2Caso1(l) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso1(1, l) - FlechasCaso1(j + 1, l)) / 100

' Else

MOM2Caso1(l) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso1(1, l) - FlechasCaso1(j, l)) / 100

MOM2Caso2(l) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso2(1, l) - FlechasCaso2(j, l)) / 100

'End If

'For X = j - 1 To NinC(1) + 1 Step -1

For X = 1 To j

MOM2Caso1(l) = MOM2Caso1(l) + Peso(X) * Dat(10) * (FlechasCaso1(X, l) - FlechasCaso1(j, l))

MOM2Caso2(l) = MOM2Caso2(l) + Peso(X) * Dat(10) * (FlechasCaso2(X, l) - FlechasCaso2(j, l))

Next

Next

End

1.1.12 Función cálculo de base

```

Sub CalculoBase()
Dim i As Integer
Dim j As Integer
Dim Peso As Double
Dim Altura As Double
Dim RPeso As Double
Dim RMomentoVuelco(2) As Double

'Puesta a 0 de las variables
For i = 1 To 2
    EsfCortante(i) = 0
    MomentoTotal(i) = 0
Next

Peso = 0
Altura = 0

'Cálculo del peso y altura de la estructura
For i = 1 To Dat(1)
    Altura = Altura + Tramo(i, 3)
    Peso = Peso + Tramo(i, 5)
Next

Peso = Peso + Dat(2) + Dat(3)

For i = 1 To NInc(2)
    EsfCortante(1) = EsfCortante(1) + FVientoCaso1(i)      'Dirección paralelo
    MomentoTotal(1) = MomentoTotal(1) + FVientoCaso1(i) * AltViento(i) 'Dirección paralelo
    EsfCortante(2) = EsfCortante(2) + FVientoCaso1(i)      'Dirección diagonal
    MomentoTotal(2) = MomentoTotal(2) + FVientoCaso1(i) * AltViento(i) 'Dirección diagonal
Next

MomentoTotal(1) = MomentoTotal(1) + MomentoSecundarioTotal(1) + MomentoPeso
MomentoTotal(2) = MomentoTotal(2) + MomentoSecundarioTotal(2) + MomentoPeso

'Cálculo de las fuerzas en los montantes debido al peso de la estructura
RPeso = Peso / 4

'Caso 1
'Cálculo de las fuerzas en los montantes debido al momento flector
RMomentoVuelco(1) = MomentoTotal(1) / (Ancho * 2)

'A continuación las sumaremos teniendo en cuenta si son de compresión o tracción
Resultados(1) = RPeso + RMomentoVuelco(1) 'Reacciones en 1 y 3
Resultados(2) = RPeso - RMomentoVuelco(1) 'Reacciones en 2 y 4

'Caso 2
'Cálculo de las fuerzas en los montantes debido al momento flector
RMomentoVuelco(2) = MomentoTotal(2) / (Ancho * Sqr(2))
  
```

'A continuación las sumaremos teniendo en cuenta si son de compresión o tracción

Resultados(3) = RPeso + RMomentoVuelco(2) 'Reacciones en 1

Resultados(4) = RPeso 'Reacciones en 2 y 3

Resultados(5) = RPeso - RMomentoVuelco(2) 'Reacciones en 4

'Momentos de Vuelco

Resultados(6) = RMomentoVuelco(1) 'Dirección Horizontal. Caso 1

Resultados(7) = RMomentoVuelco(2) 'Dirección Diagonal. Caso 2

'Esfuerzos Cortantes

Resultados(8) = EsfCortante(1) 'Dirección Horizontal. Caso 1

Resultados(9) = EsfCortante(2) 'Dirección Diagonal. Caso 2

'Reacciones Máximas

Resultados(10) = Resultados(1) 'Dirección Horizontal. Caso 1

Resultados(11) = Resultados(3) 'Dirección Diagonal. Caso 2

'Peso Estructura

Resultados(12) = Peso

'Altura Estructura

Resultados(13) = Altura

'Asignaremos los resultados obtenidos a cada cuadro de texto en el formulario de resultados

'Reacciones Caso 1

frmcalculobase.Text1(1) = Resultados(1)

frmcalculobase.Text1(2) = Resultados(2)

frmcalculobase.Text1(3) = Resultados(1)

frmcalculobase.Text1(4) = Resultados(2)

'Reacciones Caso 2

frmcalculobase.Text1(5) = Resultados(3)

frmcalculobase.Text1(6) = Resultados(4)

frmcalculobase.Text1(7) = Resultados(4)

frmcalculobase.Text1(8) = Resultados(5)

'Momentos, Reacciones Horizontales, Reacciones Máximas

frmcalculobase.Text1(9) = Resultados(6)

frmcalculobase.Text1(10) = Resultados(7)

frmcalculobase.Text1(11) = Resultados(8)

frmcalculobase.Text1(12) = Resultados(9)

frmcalculobase.Text1(13) = Resultados(10)

frmcalculobase.Text1(14) = Resultados(11)

frmcalculobase.Label3(0) = Ancho

End Sub

1.1.13 Cálculo pequeño estudio de cimentación

1.2 MÓDULO FUNCIONES

1.2.1 Declaración de variables

```
Option Explicit
Public RutaFichero As String
Public RutaBaseDatos As String
Public DirOriginal As String
Public NumTorre As Integer
Public DbBaseDatos As Database
Public RsTorre As Recordset
Public nFic As Long
Public lineatexto As String
Public CodigoTramos(80) As String
Public ContadorTramos As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer
```

1.2.2 Función abrir archivos de texto

```
Sub AbrirTexto()
Dim lineasdatos(1000) As String
Dim lineastorre(1000) As String
Dim nlineas As Integer
Dim Pos(1000) As Integer
Dim PosTramos As Integer
Dim cont As Integer

nFic = FreeFile
nlineas = 0
PosTramos = 0

Open RutaFichero For Input As nFic
    While Not EOF(1)
        Line Input #nFic, lineatexto
        nlineas = nlineas + 1
    Wend
Close #nFic

Open RutaFichero For Input As nFic
'Repite el bucle hasta el final del archivo.
While Not EOF(1)
    For i = 1 To nlineas
        If i > 500 Then GoTo salimosbucle
```

```

'Lee el carácter en la variable.
Line Input #nFic, lineatexto
lineasdatos(i) = lineatexto
If PosTrafos <> 0 Then GoTo sigue
Pos(i) = InStrRev(lineasdatos(i), "=")
PosTrafos = InStr(lineasdatos(i), "Código")
If PosTrafos <> 0 Then
    PosTrafos = i
End If

sigue:

    Next
Wend

salimosbucle:

cont = 0

    For i = 1 To nlineas
        If Not IsNumeric(Mid(lineasdatos(i), Pos(i) + 1, 7)) Then
            cont = cont + 1
        End If

        If IsNumeric(Mid(lineasdatos(i), Pos(i) + 1, 7)) Then
            lineasdatos(i - cont) = Mid(lineasdatos(i), Pos(i) + 1, 7)
        End If
    Next
Close #nFic

For i = 1 To 13
    lineasdatos(i) = Replace(lineasdatos(i), ",", ".")
    frmdatos.Text1(i) = Replace(Val(lineasdatos(i)), ",", ".")
Next

For i = 1 To Val(frmdatos.Text1(1))
    lineastorre(i) = lineasdatos(PosTrafos + i + 2)
    lineastorre(i) = Left(lineastorre(i), 10)
    lineastorre(i) = Trim(lineastorre(i))
Next

'RutaBaseDatos = direccionbase
If RutaBaseDatos <> "" Then
    Call AbrirBase
Else
    MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical,
"Aerogenerador"
    RutaBaseDatos = DirOriginal
End If

```



```
'Insertamos los tramos de torre llamando al procedimiento InsertarDatosTorre
For k = 1 To Val(frmdatos.Text1(1))
    ContadorTramos = k
    CodigoTramos(ContadorTramos) = lineastorre(k)
    frmdatos.Grid1.row = k
    Call InsertarDatosTorre
Next

frmdatos.Text1(1) = Val(frmdatos.Text1(1)) / 2
For i = Val(frmdatos.Text1(1)) + 1 To frmdatos.Grid1.Rows - 1
    frmdatos.Grid1.TextMatrix(i, 1) = ""
Next

frmdatos.cmdcalculotorre.SetFocus

End Sub
```

1.2.3 Función abrir archivos de Excel

```
Sub AbrirExcel()
    Dim apli As Excel.Application
    Dim direccionbase As String
    Dim posbase As Integer
    'Creo el objeto
    Set apli = New Excel.Application
    'Abro el archivo seleccionado la hoja y celda deseada
    apli.Workbooks.Open RutaFichero
    apli.Sheets(1).Select
    apli.Range("b3").Select

    For i = 1 To 10
        frmdatos.Text1(i).Text = apli.ActiveCell.Offset(i, 0).Value
    Next

    apli.Range("b17").Select
    For i = 1 To 3
        frmdatos.Text1(i + 10).Text = apli.ActiveCell.Offset(i, 0).Value
    Next

    For i = 1 To 13
        frmdatos.Text1(i) = Replace(frmdatos.Text1(i), ",", ".")
    Next

    apli.Range("e5").Select
    For i = 1 To Val(frmdatos.Text1(1))
        CodigoTramos(i) = apli.ActiveCell.Offset(i, 0)
    Next

    direccionbase = apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 3)
```

```

posbase = InStrRev(direccionbase, "BASE DE DATOS: ")
direccionbase = Mid(direccionbase, posbase + 15)
RutaBaseDatos = direccionbase

If RutaBaseDatos <> "" Then
    'Call AbrirBase
Else
    MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical,
    "Aerogenerador"
    RutaBaseDatos = DirOriginal
End If

'Insertamos los tramos de torre llamando al procedimiento InsertarDatosTorre
For k = 1 To Val(frmdatos.Text1(1))
    ContadorTramos = k
    If k > frmdatos.Grid1.Rows - 2 Then
        MsgBox ("Se ha producido un error al insertar los datos de torre"), vbCritical, "Aerogenerador"
        GoTo final:
    End If
    CodigoTramos(ContadorTramos) = CodigoTramos(k)
    frmdatos.Grid1.row = k
    Call InsertarDatosTorre
Next

frmdatos.Text1(1) = Val(frmdatos.Text1(1)) / 2
For i = Val(frmdatos.Text1(1)) + 1 To frmdatos.Grid1.Rows - 1
    frmdatos.Grid1.TextMatrix(i, 1) = ""
Next

final:

apli.Workbooks.Close
apli.DisplayAlerts = False
apli.Quit

frmdatos.cmdcalculotorre.SetFocus
RutaBaseDatos = direccionbase

End Sub

```

1.2.4 Función para insertar los datos de torre de la base de datos en la tabla del formulario principal

```

Sub InsertarDatosTorre()
    Dim comilla As String
    Dim criterio As String

```

Javier Vergara

```

Dim codigo As String
Dim marca As String
Dim row As Integer

comilla = Chr(34)
marca = RsTorre.Bookmark

If frmdatos.Grid1.row = 0 Then GoTo final

If CodigoTramos(ContadorTramos) = "" Then
    MsgBox "Introduzca el Código del tramo de torre", vbExclamation, "Aerogenerador"
    frmdatos.Grid1.row = frmdatos.Grid1.row - 1
    GoTo final
End If

If NumTorre < 80 Then
    codigo = CodigoTramos(ContadorTramos)
    criterio = "Código like" & comilla & codigo & comilla

    RsTorre.MoveFirst
    RsTorre.FindFirst criterio
    If RsTorre.NoMatch Then
        MsgBox "No se encontró el tipo de tramo especificado. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
        RsTorre.Bookmark = marca
        frmdatos.Grid1.row = frmdatos.Grid1.row - 1
        GoTo final
    Else
        row = frmdatos.Grid1.row
        If row = 0 Then row = 1
        frmdatos.Grid1.AddItem "", row
        For j = 1 To 9
            frmdatos.Grid1.TextMatrix(row, j + 1) = RsTorre(j - 1)
        Next
        frmdatos.Grid1.TextMatrix(row, 1) = row
        NumTorre = NumTorre + 1
        frmdatos.Text1(1) = Val(frmdatos.Text1(1)) + 1
        If NumTorre > 8 Then
            frmdatos.Grid1.Rows = NumTorre + 4
        End If
    End If
Else
    MsgBox "No se pueden introducir más tramos de torre.", vbCritical, "Aerogenerador"
    frmdatos.Text2.Text = ""
End If

final:

```

```
RsTorre.MoveFirst
For i = 1 To Val(frmdatos.Text1(1))
    frmdatos.Grid1.TextMatrix(i, 1) = i
Next
End Sub
```

1.2.5 Función para abrir la base de datos con las direcciones que obtenemos de los archivos que abrimos.

```
Sub AbrirBase()
Dim NombreTabla As String
Dim Pos As Integer
On Error GoTo final1
RsTorre.Close
DbBaseDatos.Close

Pos = InStrRev(RutaBaseDatos, ".mdb")

If Pos <> 0 Then
    Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(RutaBaseDatos)
Else
    Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(RutaBaseDatos + "\BaseDatos.mdb")
End If

NombreTabla = "Torre"
If NombreTabla = "" Then GoTo final1
Set RsTorre = DbBaseDatos.OpenRecordset(NombreTabla, dbOpenDynaset)
GoTo final

final1:

MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical, "Aerogenerador"

Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal + "\BaseDatos.mdb")
Set RsTorre = DbBaseDatos.OpenRecordset("Torre", dbOpenDynaset)
RutaBaseDatos = DirOriginal

final:

frmdatos.Combo.Clear
RsTorre.MoveLast
RsTorre.MoveFirst
frmdatos.Combo.AddItem " "
For i = 1 To RsTorre.RecordCount
    frmdatos.Combo.AddItem RsTorre(0)
    RsTorre.MoveNext
Next
```

RsTorre.MoveFirst

frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos

End Sub

1.3 MÓDULO GUARDARDATOS

1.3.1 Declaración de variables

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim Texto As String

Dim Num(30) As String

Dim Numespacio As Integer

Dim apli As Excel.Application

1.3.2 Función para escribir los datos de entrada en la caja de texto del formulario frmhojaimprimir

Sub DatosTexto()

Dim Label1(13) As String

For i = 1 To 9

Num(i) = 0 & i

Next

For i = 10 To 26

Num(i) = i

Next

Label1(1) = "	Número de Tramos....."
Label1(2) = "	Peso Buje/Caja....."
Label1(3) = "	Peso Aspas....."
Label1(4) = "	Distancia CG Buje/Caja....."
Label1(5) = "	Distancia hasta aspas....."
Label1(6) = "	Superficie Frontal Buje/Caja....."
Label1(7) = "	Altura Buje/Caja....."
Label1(8) = "	Fuerza Viento sobre aspas....."
Label1(9) = "	Viento de Referencia....."
Label1(10) = "	Coefficiente Peso Propio....."
Label1(11) = "	Densidad hormigón....."
Label1(12) = "	Tensión Admisible Terreno....."
Label1(13) = "	Altura Zapata....."

```
frmhojaimprimir.Text1 = vbCrLf & vbCrLf & Space(9) & "Fecha: " & Date & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf & Space(40) & "DATOS TORRE"
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf
```

```
For i = 1 To 10
    Texto = frmdatos.Text1(i)
    Texto = Space(7 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Label1(i) & "Dat(" & Num(i) & ")=" & Texto & vbCrLf
    If i = 1 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf
    If i = 2 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg" & vbCrLf
    If i = 3 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg" & vbCrLf
    If i = 4 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf
    If i = 5 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf
    If i = 6 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m²" & vbCrLf
    If i = 7 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf
    If i = 8 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg" & vbCrLf
    If i = 9 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m/s" & vbCrLf
    If i = 10 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf
```

Next

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf & Space(37) & "DATOS CIMENTACIÓN"
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf
```

```
For i = 11 To 13
    Texto = frmdatos.Text1(i)
    Texto = Space(7 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Label1(i) & "Dat(" & Num(i) & ")=" & Texto & vbCrLf
    If i = 11 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg/m" & vbCrLf
    If i = 12 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg/m" & vbCrLf
    If i = 13 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf
```

Next

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(38) & "TRAMOS DE TORRE" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
" & vbCrLf &
vbCrLf
```

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "      Nº  Altura  Lg Ancho  Peso |      Sección cm²  | Área SupViento
" & vbCrLf
```

```

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Código Pasos m m Kg | Montante Celo.Sup Celo.Inf | m²
" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
vbCrLf
vbCrLf
For i = 1 To NumTorre
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(1)
    j = 2
    Texto = Left(frmdatos.Grid1.TextMatrix(i, j), 9)
    Numespacio = Int((9 - Len(Texto)) / 2)
    Texto = Space(Numespacio) + Texto + Space(Numespacio)
    Texto = Space(9 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(2)
    j = 3
    Texto = frmdatos.Grid1.TextMatrix(i, j)
    Texto = Space(3 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(4)
    j = 4
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)
    Texto = Space(6 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)
    j = 5
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 3)
    Texto = Space(5 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)
    j = 6
    Texto = frmdatos.Grid1.TextMatrix(i, j)
    Texto = Space(5 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)
    j = 7
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)
    Texto = Space(6 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)
    j = 8
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)
    Texto = Space(5 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)
    j = 9
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)
    Texto = Space(5 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(8)
    j = 10
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)
    Texto = Space(5 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto

```

```

    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf
Next

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
" & vbCrLf &
vbCrLf & vbCrLf

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(38) & "BASE DE DATOS" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(10) & "Dirección: " & RutaBaseDatos & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(10) & "Tabla: " & NombreTabla & vbCrLf

End Sub

```

1.3.3 Función para escribir los resultados de base y del pequeño estudio de cimentación en la caja de texto del formulario frmhojaimprimir

```

Sub ResultadosBaseTexto()
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf & Space(34) & "CÁLCULOS BASE" & vbCrLf & vbCrLf
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(35) & "Reacciones" & vbCrLf & vbCrLf & vbCrLf
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "1      2      3      4      " & vbCrLf
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " _____ "
    & vbCrLf & vbCrLf

    For i = 1 To 2
        If i = 1 Then
            frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(4) & "Caso 1"
            For j = 1 To 4
                Texto = frmcalculobase.Text1(j)
                Texto = Space(7 - Len(Texto)) + Texto
                frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(6)
            Next
            frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf
        End If
        If i = 2 Then
            frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(4) & "Caso 2"
            For j = 5 To 8
                Texto = frmcalculobase.Text1(j)
                Texto = Space(7 - Len(Texto)) + Texto
                frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(6)
            Next
        End If
    Next

    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " _____ "
    & vbCrLf & vbCrLf

```



```

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(36) & "CASO 1" & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Par de Vuelco....."
Texto = frmcalculobase.Text1(9)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg·m" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Reacción Horizontal....."
Texto = frmcalculobase.Text1(11)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Reacción Máxima....."
Texto = frmcalculobase.Text1(13)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg" & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(36) & "CASO 2" & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Par de Vuelco....."
Texto = frmcalculobase.Text1(10)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg·m" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Reacción Horizontal....."
Texto = frmcalculobase.Text1(12)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Reacción Máxima....."
Texto = frmcalculobase.Text1(14)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg" & vbCrLf & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(30) & "CÁLCULOS CIMENTACIÓN" & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Longitud Zapata....."
Texto = frmcalculobase.Text1(15)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " m" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Altura Zapata....."
Texto = frmcalculobase.Text1(16)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " m" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Tensión Zapata....."
Texto = frmcalculobase.Text1(17)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg/m²" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Distribución de Tensiones."
Texto = frmcalculobase.Text1(18)
Texto = Space(9 - Len(Texto)) + Texto
rmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & vbCrLf

End Sub

```

1.3.4 Función para escribir los resultados de torre en la caja de texto del formulario frmhojaimprimir

Sub ResultadosTorreTexto()

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(37) & "CÁLCULO DE TORRE"
```

```
For k = 1 To 2
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf
```

```
    If k = 1 Then
```

```
        frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(32) & "CASO 1 - DIRECCIÓN HORIZONTAL" &
```

```
vbCrLf & vbCrLf
```

```
    Else
```

```
        frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(33) & "CASO 2 - DIRECCIÓN DIAGONAL" &
```

```
vbCrLf & vbCrLf
```

```
    End If
```

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
```

```
_____ " & vbCrLf &
```

```
vbCrLf
```

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "  Nº  Altura  Peso  Esfuerzo  Momento  Flecha |  Tensión - Kg/cm²
```

```
| " & vbCrLf
```

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Tramo    m    Kg  Cortante Kg  Kg-m    cm |Montante  Celosía Sup
```

```
Celosía Inf| " & vbCrLf
```

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
```

```
_____ " & vbCrLf &
```

```
vbCrLf
```

```
For i = 1 To Dat(1)
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(3)
```

```
    j = 1
```

```
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
```

```
    Texto = Space(2 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
```

```
    j = 2
```

```
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
```

```
    Texto = Space(3 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
```

```
    j = 3
```

```
    Texto = Left(frmcalculotorre.Grid1(k).TextMatrix(i, j), 9)
```

```
    Numespacio = Int((9 - Len(Texto)) / 2)
```

```
    Texto = Space(Numespacio) + Texto + Space(Numespacio)
```

```
    Texto = Space(9 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(1)
```

```
    j = 4
```

```
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
```

```
    Texto = Space(7 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(2)
```

```

j = 5
Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
Texto = Space(6 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)

j = 6
Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
Texto = Space(6 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)

j = 7
Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
Texto = Space(6 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)

j = 8
Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
Texto = Space(5 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(4)

j = 9
Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
Texto = Space(5 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf

Next
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
" & vbCrLf &
vbCrLf
Next

End Sub

```

1.3.5 Función para escribir tanto los datos como todos los resultados en la caja de texto del formulario frmhojaimprimir

```

Sub TodoResultadosTexto()
Call ResultadosTorreTexto
Call ResultadosBaseTexto
End Sub

```

1.3.6 Función para escribir los datos de entrada en una hoja de Excel

```

Sub DatosExcel()
Clipboard.Clear
Set apli = New Excel.Application
apli.Workbooks.Add
apli.Worksheets(1).Range("b1").Select
apli.WindowState = xlMaximized

```

```

apli.Visible = True
apli.ErrorCheckingOptions.BackgroundChecking = False
apli.Sheets("hoja1").Name = "Datos Torre"
apli.Columns(1).ColumnWidth = 29.29
apli.Columns(13).ColumnWidth = 14.57
apli.Range("a2").HorizontalAlignment = xlCenter
apli.Range("a2").Font.Bold = True

apli.Range("a2") = "DATOS TORRE"
apli.Range("a16").HorizontalAlignment = xlCenter
apli.Range("a16") = "DATOS CIMENTACIÓN"
apli.Range("a16").Font.Bold = True

apli.Range("a3").Select
For i = 1 To 10
    apli.ActiveCell.Offset(i, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i, 1).Value = Val(frmdatos.Text1(i))
Next

apli.Range("a17").Select
For i = 11 To 13
    apli.ActiveCell.Offset(i - 10, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i - 10, 1).Value = Val(frmdatos.Text1(i))
Next

apli.Columns("D:M").HorizontalAlignment = xlCenter
apli.Columns("D:M").NumberFormat = "@"
apli.Range("g2") = "TRAMOS DE TORRE"
apli.Range("g2").Font.Bold = True
apli.Range("g2", "i2").Merge
apli.Range("d4") = "Nº"
apli.Range("f4") = "Nº"
apli.Range("g4") = "Altura"
apli.Range("h4") = "Lg Ancho"
apli.Range("i4") = "Peso"
apli.Range("j4") = "Sección cm²"
apli.Range("j4", "l4").Merge
apli.Range("m4") = "Área Sup Viento"
apli.Range("d5") = "Tramo"
apli.Range("e5") = "Código"
apli.Range("f5") = "Pasos"
apli.Range("g5") = "m"
apli.Range("h5") = "m"
apli.Range("i5") = "Kg"
apli.Range("j5") = "Montante"
apli.Range("k5") = "Celosía Sup"

```

```

apli.Range("l5") = "Celosía Inf"
apli.Range("m5") = "Tramo m2"

apli.Range("d6").Select

Clipboard.Clear
frmdatos.Grid1.Col = 1
frmdatos.Grid1.Row = 1
frmdatos.Grid1.ColSel = 10
frmdatos.Grid1.RowSel = Val(frmdatos.Text1(1)) + 1
Clipboard.SetText frmdatos.Grid1.Clip
apli.ActiveSheet.Paste
frmdatos.Grid1.ColSel = 1
frmdatos.Grid1.RowSel = 1
Clipboard.Clear

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0) = "BASE DE DATOS"
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0), apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 1)).Merge
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0).HorizontalAlignment = xlLeft
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0) = "Dirección: " & RutaBaseDatos
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, 0) = "Tabla: " & NombreTabla

apli.Range("b1").Value = Date
apli.Range("a1").Select
et apli = Nothing

End Sub

```

1.3.7 Función para escribir tanto los datos como los resultados de base y del pequeño estudio de cimentación en una hoja de Excel

```

Sub ResultadosBaseExcel()
Clipboard.Clear
Set apli = New Excel.Application
apli.Workbooks.Add
apli.Worksheets(1).Range("b1").Select
apli.WindowState = xlMaximized
apli.Visible = True
apli.ErrorCheckingOptions.BackgroundChecking = False
apli.Sheets("hoja1").Name = "Datos Torre"
apli.Sheets("hoja2").Name = "Cálculos Base"

'CÁLCULOS BASE/CIMENTACIÓN

apli.Sheets("Cálculos Base").Select
apli.Columns("A:I").HorizontalAlignment = xlCenter

```

```

apli.Columns(6).ColumnWidth = 17
apli.Columns(8).ColumnWidth = 17
apli.Range("c3") = "CÁLCULOS BASE"
apli.Range("c3").Font.Bold = True
apli.Range("c3", "d3").Merge
apli.Range("c6") = "Reacciones"
apli.Range("c6").Font.Bold = True
apli.Range("c6", "d6").Merge
apli.Range("f6") = "Caso 1"
apli.Range("f6").Font.Bold = True
apli.Range("f6", "g6").Merge
apli.Range("h6") = "Caso 2"
apli.Range("h6").Font.Bold = True
apli.Range("h6", "i6").Merge
apli.Range("a8") = "Caso 1"
apli.Range("a10") = "Caso 2"
apli.Range("f8") = "Par de Vuelco"
apli.Range("f9") = "Reacción Horizontal"
apli.Range("f10") = "Reacción Máxima"
apli.Range("h8") = "Par de Vuelco"
apli.Range("h9") = "Reacción Horizontal"
apli.Range("h10") = "Reacción Máxima"

apli.Range("b7") = 1
apli.Range("c7") = 2
apli.Range("d7") = 3
apli.Range("e7") = 4
apli.Range("c13") = "CÁLCULO CIMENTACIÓN"
apli.Range("c13").Font.Bold = True
apli.Range("c13", "d13").Merge
apli.Range("b15") = "Longitud Zapata"
apli.Range("b15", "c15").Merge
apli.Range("b16") = "Altura Zapata"
apli.Range("b16", "c16").Merge
apli.Range("b17") = "Tensión Zapata"
apli.Range("b17", "c17").Merge
apli.Range("b18") = "Distribución Tensiones"
apli.Range("b18", "c18").Merge

apli.Range("b8").Select
For i = 1 To 4
    apli.ActiveCell.Offset(0, i - 1) = Val(frmcalculobase.Text1(i))
    apli.ActiveCell.Offset(2, i - 1) = Val(frmcalculobase.Text1(i * 2))
Next
apli.Range("g8") = Val(frmcalculobase.Text1(9))
apli.Range("g9") = Val(frmcalculobase.Text1(11))
apli.Range("g10") = Val(frmcalculobase.Text1(13))
  
```



```

apli.Range("i8") = Val(frmcalculobase.Text1(10))
apli.Range("i9") = Val(frmcalculobase.Text1(12))
apli.Range("i10") = Val(frmcalculobase.Text1(14))
apli.Range("d15") = Val(frmcalculobase.Text1(15))
apli.Range("d16") = Val(frmcalculobase.Text1(16))
apli.Range("d17") = Val(frmcalculobase.Text1(17))
apli.Range("d18") = Val(frmcalculobase.Text1(1))
apli.Range("a1") = Date

```

```

apli.Range("a1").Select

```

'DATOS DE TORRE

```

apli.Sheets("Datos Torre").Select
apli.Columns(1).ColumnWidth = 29.29
apli.Columns(13).ColumnWidth = 14.57
apli.Range("a2").HorizontalAlignment = xlCenter
apli.Range("a2").Font.Bold = True
apli.Range("a2") = "DATOS TORRE"
apli.Range("a16").HorizontalAlignment = xlCenter
apli.Range("a16") = "DATOS CIMENTACIÓN"
apli.Range("a16").Font.Bold = True

```

```

apli.Range("a3").Select
For i = 1 To 10
    apli.ActiveCell.Offset(i, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i, 1).Value = Val(frmdatos.Text1(i))

```

Next

```

apli.Range("a17").Select
For i = 11 To 13
    apli.ActiveCell.Offset(i - 10, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i - 10, 1).Value = Val(frmdatos.Text1(i))

```

Next

```

apli.Columns("D:M").HorizontalAlignment = xlCenter
apli.Columns("D:M").NumberFormat = "@"
apli.Range("g2") = "TRAMOS DE TORRE"
apli.Range("g2").Font.Bold = True
apli.Range("g2", "i2").Merge
apli.Range("d4") = "Nº"
apli.Range("f4") = "Nº"
apli.Range("g4") = "Altura"
apli.Range("h4") = "Lg Ancho"
apli.Range("i4") = "Peso"
apli.Range("j4") = "Sección cm²"

```



```

apli.Range("j4", "l4").Merge
apli.Range("m4") = "Área Sup Viento"
apli.Range("d5") = "Tramo"
apli.Range("e5") = "Código"
apli.Range("f5") = "Pasos"
apli.Range("g5") = "m"
apli.Range("h5") = "m"
apli.Range("i5") = "Kg"
apli.Range("j5") = "Montante"
apli.Range("k5") = "Celosía Sup"
apli.Range("l5") = "Celosía Inf"
apli.Range("m5") = "Tramo m²"

apli.Range("d6").Select

Clipboard.Clear
frmdatos.Grid1.Col = 1
frmdatos.Grid1.row = 1
frmdatos.Grid1.ColSel = 10
frmdatos.Grid1.RowSel = Val(frmdatos.Text1(1)) + 1
Clipboard.SetText frmdatos.Grid1.Clip
apli.ActiveSheet.Paste
frmdatos.Grid1.ColSel = 1
frmdatos.Grid1.RowSel = 1
Clipboard.Clear

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0) = "BASE DE DATOS"
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0), apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 1)).Merge
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0).HorizontalAlignment = xlLeft
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0) = "Dirección: " & RutaBaseDatos
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, 0) = "Tabla: " & NombreTabla

apli.Range("b1").Value = Date
apli.Range("a1").Select
Set apli = Nothing

End Sub

```

1.3.9 Función para escribir tanto los datos como los resultados de torre en una hoja de Excel

```

Sub ResultadosTorreExcel()
Clipboard.Clear
Set apli = New Excel.Application
apli.Workbooks.Add

```

```

apli.Worksheets(1).Range("b1").Select
apli.WindowState = xlMaximized
apli.Visible = True
apli.ErrorCheckingOptions.BackgroundChecking = False
apli.Sheets("hoja1").Name = "Datos Torre"
apli.Sheets("hoja2").Name = "Cálculos Torre"

```

'CÁLCULOS TORRE

```

Dim numfila As Integer
numfila = Dat(1)
apli.Sheets("Cálculos Torre").Select
apli.Columns("A:I").HorizontalAlignment = xlCenter
apli.Columns(4).ColumnWidth = 11.71
apli.Range("c3") = "CASO 1 - DIRECCIÓN HORIZONTAL"
apli.Range("c3").Font.Bold = True
apli.Range("c3", "f3").Merge
apli.Range("a5") = "Nº"
apli.Range("b5") = "Altura"
apli.Range("c5") = "Peso"
apli.Range("d5") = "Esfuerzo"
apli.Range("e5") = "Momento"
apli.Range("f5") = "Flecha"
apli.Range("g5") = "Tensión - Kg/cm²"
apli.Range("g5", "i5").Merge
apli.Range("a6") = "Tramo"
apli.Range("b6") = "m"
apli.Range("c6") = "Kg"
apli.Range("d6") = "Cortante - Kg"
apli.Range("e6") = "Kg·m"
apli.Range("f6") = "cm"
apli.Range("g6") = "Montante"
apli.Range("h6") = "Celosía Sup"
apli.Range("i6") = "Celosía Inf"

apli.Range("a6").Select
apli.ActiveCell.Offset(numfila + 2, 2) = "CASO 2 - DIRECCIÓN DIAGONAL"
apli.ActiveCell.Offset(numfila + 2, 2).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(numfila + 2, 2), apli.ActiveCell.Offset(numfila + 2, 5)).Merge
apli.ActiveCell.Offset(numfila + 4, 0) = "Nº"
apli.ActiveCell.Offset(numfila + 4, 1) = "Altura"
apli.ActiveCell.Offset(numfila + 4, 2) = "Peso"
apli.ActiveCell.Offset(numfila + 4, 3) = "Esfuerzo"
apli.ActiveCell.Offset(numfila + 4, 4) = "Momento"
apli.ActiveCell.Offset(numfila + 4, 5) = "Flecha"
apli.ActiveCell.Offset(numfila + 4, 6) = "Tensión - Kg/cm²"
apli.Range(apli.ActiveCell.Offset(numfila + 4, 6), apli.ActiveCell.Offset(numfila + 4, 8)).Merge

```

```

apli.ActiveCell.Offset(numfila + 5, 0) = "Tramo"
apli.ActiveCell.Offset(numfila + 5, 1) = "m"
apli.ActiveCell.Offset(numfila + 5, 2) = "Kg"
apli.ActiveCell.Offset(numfila + 5, 3) = "Cortante - Kg"
apli.ActiveCell.Offset(numfila + 5, 4) = "Kg·m"
apli.ActiveCell.Offset(numfila + 5, 5) = "cm"
apli.ActiveCell.Offset(numfila + 5, 6) = "Montante"
apli.ActiveCell.Offset(numfila + 5, 7) = "Celosía Sup"
apli.ActiveCell.Offset(numfila + 5, 8) = "Celosía Inf"
apli.Range("a7").Select
Clipboard.Clear
For j = 1 To 2
    Clipboard.Clear
    frmcalculotorre.Grid1(j).Col = 1
    frmcalculotorre.Grid1(j).row = 1
    frmcalculotorre.Grid1(j).ColSel = 9
    frmcalculotorre.Grid1(j).RowSel = numfila
    Clipboard.SetText frmcalculotorre.Grid1(j).Clip
    apli.ActiveSheet.Paste
    frmcalculotorre.Grid1(j).ColSel = 1
    frmcalculotorre.Grid1(j).RowSel = 1
    Clipboard.Clear
    apli.ActiveCell.Offset(numfila + 5, 0).Select
Next

apli.Range("a1") = Date
apli.Range("a1").Select

'DATOS DE TORRE

apli.Sheets("Datos Torre").Select
apli.Columns(1).ColumnWidth = 29.29
apli.Columns(13).ColumnWidth = 14.57
apli.Range("a2").HorizontalAlignment = xlCenter
apli.Range("a2").Font.Bold = True
apli.Range("a2") = "DATOS TORRE"
apli.Range("a16").HorizontalAlignment = xlCenter
apli.Range("a16") = "DATOS CIMENTACIÓN"
apli.Range("a16").Font.Bold = True

apli.Range("a3").Select
For i = 1 To 10
    apli.ActiveCell.Offset(i, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i, 1).Value = Val(frmdatos.Text1(i))
Next

apli.Range("a17").Select
For i = 11 To 13

```

```
apli.ActiveCell.Offset(i - 10, 0) = frmdatos.Label1(i)
apli.ActiveCell.Offset(i - 10, 1).Value = Val(frmdatos.Text1(i))
```

Next

```
apli.Columns("D:M").HorizontalAlignment = xlCenter
apli.Columns("D:M").NumberFormat = "@"
apli.Range("g2") = "TRAMOS DE TORRE"
apli.Range("g2").Font.Bold = True
apli.Range("g2", "i2").Merge
apli.Range("d4") = "Nº"
apli.Range("f4") = "Nº"
apli.Range("g4") = "Altura"
apli.Range("h4") = "Lg Ancho"
apli.Range("i4") = "Peso"
apli.Range("j4") = "Sección cm²"
apli.Range("j4", "l4").Merge
apli.Range("m4") = "Área Sup Viento"
apli.Range("d5") = "Tramo"
apli.Range("e5") = "Código"
apli.Range("f5") = "Pasos"
apli.Range("g5") = "m"
apli.Range("h5") = "m"
apli.Range("i5") = "Kg"
apli.Range("j5") = "Montante"
apli.Range("k5") = "Celosía Sup"
apli.Range("l5") = "Celosía Inf"
apli.Range("m5") = "Tramo m²"

apli.Range("d6").Select
Clipboard.Clear
frmdatos.Grid1.Col = 1
frmdatos.Grid1.Row = 1
frmdatos.Grid1.ColSel = 10
frmdatos.Grid1.RowSel = Val(frmdatos.Text1(1)) + 1
Clipboard.SetText frmdatos.Grid1.Clip
apli.ActiveSheet.Paste
frmdatos.Grid1.ColSel = 1
frmdatos.Grid1.RowSel = 1
Clipboard.Clear

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0) = "BASE DE DATOS"

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0), apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 1)).Merge
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0).HorizontalAlignment = xlLeft
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0) = "Dirección: " & RutaBaseDatos
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, 0) = "Tabla: " & NombreTabla
```

```
apli.Range("b1").Value = Date
apli.Range("a1").Select
Set apli = Nothing
```

```
End Sub
```

1.3.10 Función para escribir tanto los datos como todos los resultados en una hoja de Excel

```
Sub TodoResultadosExcel()
Clipboard.Clear
Set apli = New Excel.Application
apli.Workbooks.Add
apli.Worksheets(1).Range("b1").Select
apli.WindowState = xlMaximized
apli.Visible = True
apli.ErrorCheckingOptions.BackgroundChecking = False
apli.Sheets("hoja1").Name = "Datos Torre"
apli.Sheets("hoja2").Name = "Cálculos Torre"
apli.Sheets("hoja3").Name = "Cálculos Base"
```

```
'CÁLCULOS BASE/CIMENTACIÓN
```

```
apli.Sheets("Cálculos Base").Select
apli.Columns("A:I").HorizontalAlignment = xlCenter
apli.Columns(6).ColumnWidth = 17
apli.Columns(8).ColumnWidth = 17
apli.Range("c3") = "CÁLCULOS BASE"
apli.Range("c3").Font.Bold = True
apli.Range("c3", "d3").Merge
apli.Range("c6") = "Reacciones"
apli.Range("c6").Font.Bold = True
apli.Range("c6", "d6").Merge
apli.Range("f6") = "Caso 1"
apli.Range("f6").Font.Bold = True
apli.Range("f6", "g6").Merge
apli.Range("h6") = "Caso 2"
apli.Range("h6").Font.Bold = True
apli.Range("h6", "i6").Merge
apli.Range("a8") = "Caso 1"
apli.Range("a10") = "Caso 2"
apli.Range("f8") = "Par de Vuelco"
apli.Range("f9") = "Reacción Horizontal"
apli.Range("f10") = "Reacción Máxima"
apli.Range("h8") = "Par de Vuelco"
apli.Range("h9") = "Reacción Horizontal"
```

```

apli.Range("h10") = "Reacción Máxima"
apli.Range("b7") = 1
apli.Range("c7") = 2
apli.Range("d7") = 3
apli.Range("e7") = 4
apli.Range("c13") = "CÁLCULO CIMENTACIÓN"
apli.Range("c13").Font.Bold = True
apli.Range("c13", "d13").Merge
apli.Range("b15") = "Longitud Zapata"
apli.Range("b15", "c15").Merge
apli.Range("b16") = "Altura Zapata"
apli.Range("b16", "c16").Merge
apli.Range("b17") = "Tensión Zapata"
apli.Range("b17", "c17").Merge
apli.Range("b18") = "Distribución Tensiones"
apli.Range("b18", "c18").Merge

apli.Range("b8").Select
For i = 1 To 4
    apli.ActiveCell.Offset(0, i - 1) = Val(frmcalculobase.Text1(i))
    apli.ActiveCell.Offset(2, i - 1) = Val(frmcalculobase.Text1(i * 2))
Next

apli.Range("g8") = Val(frmcalculobase.Text1(9))
apli.Range("g9") = Val(frmcalculobase.Text1(11))
apli.Range("g10") = Val(frmcalculobase.Text1(13))
apli.Range("i8") = Val(frmcalculobase.Text1(10))
apli.Range("i9") = Val(frmcalculobase.Text1(12))
apli.Range("i10") = Val(frmcalculobase.Text1(14))
apli.Range("d15") = Val(frmcalculobase.Text1(15))
apli.Range("d16") = Val(frmcalculobase.Text1(16))
apli.Range("d17") = Val(frmcalculobase.Text1(17))
apli.Range("d18") = Val(frmcalculobase.Text1(1))

apli.Range("a1") = Date
apli.Range("a1").Select

'CÁLCULOS TORRE

Dim numfila As Integer
numfila = Dat(1)
apli.Sheets("Cálculos Torre").Select
apli.Columns("A:I").HorizontalAlignment = xlCenter
apli.Columns(4).ColumnWidth = 11.71
apli.Range("c3") = "CASO 1 - DIRECCIÓN HORIZONTAL"
apli.Range("c3").Font.Bold = True
apli.Range("c3", "f3").Merge

```

```

apli.Range("a5") = "Nº"
apli.Range("b5") = "Altura"
apli.Range("c5") = "Peso"
apli.Range("d5") = "Esfuerzo"
apli.Range("e5") = "Momento"
apli.Range("f5") = "Flecha"
apli.Range("g5") = "Tensión - Kg/cm²"
apli.Range("g5", "i5").Merge
apli.Range("a6") = "Tramo"
apli.Range("b6") = "m"
apli.Range("c6") = "Kg"
apli.Range("d6") = "Cortante - Kg"
apli.Range("e6") = "Kg·m"
apli.Range("f6") = "cm"
apli.Range("g6") = "Montante"
apli.Range("h6") = "Celosía Sup"
apli.Range("i6") = "Celosía Inf"

apli.Range("a6").Select
apli.ActiveCell.Offset(numfila + 2, 2) = "CASO 2 - DIRECCIÓN DIAGONAL"
apli.ActiveCell.Offset(numfila + 2, 2).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(numfila + 2, 2), apli.ActiveCell.Offset(numfila + 2, 5)).Merge
apli.ActiveCell.Offset(numfila + 4, 0) = "Nº"
apli.ActiveCell.Offset(numfila + 4, 1) = "Altura"
apli.ActiveCell.Offset(numfila + 4, 2) = "Peso"
apli.ActiveCell.Offset(numfila + 4, 3) = "Esfuerzo"
apli.ActiveCell.Offset(numfila + 4, 4) = "Momento"
apli.ActiveCell.Offset(numfila + 4, 5) = "Flecha"
apli.ActiveCell.Offset(numfila + 4, 6) = "Tensión - Kg/cm²"
apli.Range(apli.ActiveCell.Offset(numfila + 4, 6), apli.ActiveCell.Offset(numfila + 4, 8)).Merge
apli.ActiveCell.Offset(numfila + 5, 0) = "Tramo"
apli.ActiveCell.Offset(numfila + 5, 1) = "m"
apli.ActiveCell.Offset(numfila + 5, 2) = "Kg"
apli.ActiveCell.Offset(numfila + 5, 3) = "Cortante - Kg"
apli.ActiveCell.Offset(numfila + 5, 4) = "Kg·m"
apli.ActiveCell.Offset(numfila + 5, 5) = "cm"
apli.ActiveCell.Offset(numfila + 5, 6) = "Montante"
apli.ActiveCell.Offset(numfila + 5, 7) = "Celosía Sup"
apli.ActiveCell.Offset(numfila + 5, 8) = "Celosía Inf"

apli.Range("a7").Select
Clipboard.Clear
For j = 1 To 2
    Clipboard.Clear
    frmcalculotorre.Grid1(j).Col = 1
    frmcalculotorre.Grid1(j).row = 1
    frmcalculotorre.Grid1(j).ColSel = 9

```

```
frmcalculotorre.Grid1(j).RowSel = numfila
Clipboard.SetText frmcalculotorre.Grid1(j).Clip
apli.ActiveSheet.Paste
frmcalculotorre.Grid1(j).ColSel = 1
frmcalculotorre.Grid1(j).RowSel = 1
Clipboard.Clear
apli.ActiveCell.Offset(numfila + 5, 0).Select
```

Next

```
apli.Range("a1") = Date
apli.Range("a1").Select
```

'DATOS DE TORRE

```
apli.Sheets("Datos Torre").Select
apli.Columns(1).ColumnWidth = 29.29
apli.Columns(13).ColumnWidth = 14.57
apli.Range("a2").HorizontalAlignment = xlCenter
apli.Range("a2").Font.Bold = True
apli.Range("a2") = "DATOS TORRE"
apli.Range("a16").HorizontalAlignment = xlCenter
apli.Range("a16") = "DATOS CIMENTACIÓN"
apli.Range("a16").Font.Bold = True
```

```
apli.Range("a3").Select
For i = 1 To 10
    apli.ActiveCell.Offset(i, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i, 1).Value = Val(frmdatos.Text1(i))
```

Next

```
apli.Range("a17").Select
For i = 11 To 13
    apli.ActiveCell.Offset(i - 10, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i - 10, 1).Value = Val(frmdatos.Text1(i))
```

Next

```
apli.Columns("D:M").HorizontalAlignment = xlCenter
apli.Columns("D:M").NumberFormat = "@"
apli.Range("g2") = "TRAMOS DE TORRE"
apli.Range("g2").Font.Bold = True
apli.Range("g2", "i2").Merge
apli.Range("d4") = "Nº"
apli.Range("f4") = "Nº"
apli.Range("g4") = "Altura"
apli.Range("h4") = "Lg Ancho"
apli.Range("i4") = "Peso"
apli.Range("j4") = "Sección cm²"
apli.Range("j4", "l4").Merge
```



```

apli.Range("m4") = "Área Sup Viento"
apli.Range("d5") = "Tramo"
apli.Range("e5") = "Código"
apli.Range("f5") = "Pasos"
apli.Range("g5") = "m"
apli.Range("h5") = "m"
apli.Range("i5") = "Kg"
apli.Range("j5") = "Montante"
apli.Range("k5") = "Celosía Sup"
apli.Range("l5") = "Celosía Inf"
apli.Range("m5") = "Tramo m²"

apli.Range("d6").Select
Clipboard.Clear
frmdatos.Grid1.Col = 1
frmdatos.Grid1.Row = 1
frmdatos.Grid1.ColSel = 10
frmdatos.Grid1.RowSel = Val(frmdatos.Text1(1)) + 1
Clipboard.SetText frmdatos.Grid1.Clip
apli.ActiveSheet.Paste
frmdatos.Grid1.ColSel = 1
frmdatos.Grid1.RowSel = 1
Clipboard.Clear

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0) = "BASE DE DATOS"
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0), apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 1)).Merge
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0).HorizontalAlignment = xlLeft
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0) = "Dirección: " & RutaBaseDatos
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, 0) = "Tabla: " & NombreTabla

apli.Range("b1").Value = Date
apli.Range("a1").Select
Set apli = Nothing

End Sub

```

1.4 FORMULARIO DE DATOS.

1.4.1 Eventos de formulario

Cargar el formulario

Private Sub Form_Load()

Describimos las tablas

With Grid1

```
.TextArray(1) = "Tramo"
.TextArray(2) = "Código"
.TextArray(3) = "Pasos"
.TextArray(4) = "m"
.TextArray(5) = "m"
.TextArray(6) = "Kg"
.TextArray(7) = "Montante"
.TextArray(8) = "Celosía Sup"
.TextArray(9) = "Celosía Inf"
.TextArray(10) = "Tramo m²"
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1100
.ColWidth(3) = 800
.ColWidth(4) = 900
.ColWidth(5) = 950
.ColWidth(6) = 950
.ColWidth(7) = 1000
.ColWidth(8) = 1350
.ColWidth(9) = 1350
.ColWidth(10) = 1700
```

End With

With Grid2

```
.RowHeight(0) = 450
.TextArray(1) = "Nº"
.TextArray(3) = "Nº"
.TextArray(4) = "Altura"
.TextArray(5) = "Lg Ancho"
.TextArray(6) = "Peso"
.TextArray(7) = "Sección cm²"
.TextArray(8) = "Sección cm²"
.TextArray(9) = "Sección cm²"
.TextArray(10) = "Área Sup Viento"
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1100
```

```

.ColWidth(3) = 800
.ColWidth(4) = 900
.ColWidth(5) = 950
.ColWidth(6) = 950
.ColWidth(7) = 1000
.ColWidth(8) = 1350
.ColWidth(9) = 1350
.ColWidth(10) = 1950
.MergeRow(0) = True
End With

'Damos la orientación de texto deseada a las columnas
For i = 1 To 10
    Grid1.ColAlignment(i) = 4
    Grid2.ColAlignment(i) = 4
Next

'Abrimos el texto y obtenemos la dirección de la base de datos
nFic = FreeFile
Open "BaseDatos.cfg" For Input As nFic
    Line Input #nFic, lineatexto
    Line Input #nFic, lineatexto
    i = InStr(lineatexto, Chr(34))
    j = InStr(i + 1, lineatexto, Chr(34))
    RutaBaseDatos = Mid(lineatexto, i + 1, j - i - 1)
    DirOriginal = RutaBaseDatos
Close

DirOriginal = DirOriginal + "\BaseDatos"
RutaBaseDatos = DirOriginal
TablaOriginal = "Torre"
NombreTabla = TablaOriginal

'Abrimos la base de datos de los tramos de torre
Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal)
Set RsTorre = DbBaseDatos.OpenRecordset(TablaOriginal, dbOpenDynaset)

'Llevamos todos los códigos de tramo al combo
RsTorre.MoveLast
RsTorre.MoveFirst
Combo.AddItem " "
For i = 1 To RsTorre.RecordCount
    Combo.AddItem RsTorre(0)
    RsTorre.MoveNext
Next
RsTorre.MoveFirst

```

```
'Al índice de las cajas de texto le asignamos el valor 2  
xi = 2
```

```
End Sub
```

Activar el formulario

```
Private Sub Form_Activate()  
frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos  
End Sub
```

Arrastrar algún archivo sobre el formulario

```
Private Sub Form_OLEDragDrop(Data As DataObject, Effect As Long, Button As Integer, Shift As Integer, X As Single, y As Single)  
'On Error GoTo Final
```

```
'Si no es un archivo lo que se arrastra, habrá error  
If Data.GetFormat(vbCFFiles) = False Then  
    MsgBox "No se ha arrastrado ningún archivo", vbInformation  
    Exit Sub  
End If
```

```
'Si se arrastra más de un archivo, habrá error  
If Data.Files.Count > 1 Then  
    MsgBox "No se puede arrastrar más de un archivo"  
    Exit Sub  
End If
```

```
    cmdlimpiardatos_Click  
cmdlimpiartramos_Click
```

```
Dim posnombre As Integer  
Dim Texto As String  
RutaFichero = Data.Files(1)  
posnombre = InStrRev(RutaFichero, ".")  
Texto = Mid(RutaFichero, posnombre + 1)
```

```
'Comprobamos si es un archivo Excel o de texto  
If Texto = "txt" Then  
    Call AbrirTexto  
Else  
    Call AbrirExcel  
End If  
  
final:
```

Javier Vergara

End Sub

Descargar el formulario

```
Private Sub Form_Unload(Cancel As Integer)
End
End Sub
```

1.4.2 Eventos en cajas de texto

Al pulsar cualquier tecla sobre Text1

```
Private Sub Text1_KeyPress(Index As Integer, KeyAscii As Integer)
Dim X As Integer
'Sólo se pueden teclear en los textos número y "."
If KeyAscii <> Asc("9") Then
If KeyAscii <> Asc("8") Then
If KeyAscii <> Asc("7") Then
If KeyAscii <> Asc("6") Then
If KeyAscii <> Asc("5") Then
If KeyAscii <> Asc("4") Then
If KeyAscii <> Asc("3") Then
If KeyAscii <> Asc("2") Then
If KeyAscii <> Asc("1") Then
If KeyAscii <> Asc("0") Then
If KeyAscii <> 8 Then      'tecla suprimir
    X = 1
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
If KeyAscii = Asc(".") Then
    X = 2
End If
If KeyAscii = Asc(",") Then
    X = 3
End If
If KeyAscii = vbKeyReturn Then
    X = 4
End If
```

```

If X = 1 Then
    KeyAscii = 0
    MsgBox "Introduzca sólo números", vbInformation
End If
If X = 2 Then
    KeyAscii = Asc(".")
End If
If X = 3 Then
    KeyAscii = Asc(".")
End If
If X = 4 Then
    xi = xi + 1
    If xi = 14 Then
        Combo.SetFocus
        xi = 2
        Exit Sub
    End If
    Text1(xi).SetFocus
End If

End Sub
  
```

Al hacer Click con el ratón sobre la caja de texto Text1

```

Private Sub Text1_Click(Index As Integer)
    xi = Index
End Sub
  
```

Al pulsar cualquier tecla sobre Text2

```

Private Sub Text2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        cmdinsertar_Click
    End If
End Sub
  
```

1.4.3 Eventos de botones

Botón para bajar un tramo de torre

```

Private Sub cmdbajar_Click()
    Dim tramos(10) As String
    With Grid1
  
```

```

    If Grid1.TextMatrix(.row, 2) = "" Then
        MsgBox "Seleccione un tramo de torre para desplazarlo.", vbInformation, "Aerogenerador"
        Exit Sub
    End If
    If Grid1.row = NumTorre Then Exit Sub
    For j = 2 To 10
        tramos(j) = Grid1.TextMatrix(.row + 1, j)
        Grid1.TextMatrix(.row + 1, j) = Grid1.TextMatrix(.row, j)
        Grid1.TextMatrix(.row, j) = tramos(j)
    Next
    Grid1.row = Grid1.row + 1
    Grid1.TopRow = Grid1.row - 1
    Grid1.Col = 0
    Grid1.ColSel = 10
End With

End Sub

```

Botón para subir un tramo de torre

```

Private Sub cmdsubir_Click()
    Dim tramos(10) As String
    With Grid1
        If Grid1.TextMatrix(.row, 2) = "" Then
            MsgBox "Seleccione un tramo de torre para desplazarlo.", vbInformation, "Aerogenerador"
            Exit Sub
        End If
        If Grid1.row = 1 Then Exit Sub 'GoTo final
        For j = 2 To 10
            tramos(j) = Grid1.TextMatrix(.row - 1, j)
            Grid1.TextMatrix(.row - 1, j) = Grid1.TextMatrix(.row, j)
            Grid1.TextMatrix(.row, j) = tramos(j)
        Next
        Grid1.row = Grid1.row - 1
        If Grid1.row <> 1 Then
            Grid1.TopRow = Grid1.row - 1
        End If
        Grid1.Col = 0
        Grid1.ColSel = 10
    End With

End Sub

```

Botón para eliminar un tramo de torre

```

Private Sub cmdeliminar_Click()

```



```

If Grid1.Rows > 10 Then
    With Grid1
        If Grid1.TextMatrix(.row, 2) = "" Then
            MsgBox "No hay ninguna fila seleccionada para eliminar.", vbExclamation, "Aerogenerador"
            Exit Sub
        Else
            If frmdatos.Grid1.row = 0 Then frmdatos.Grid1.row = 1
                .RemoveItem .row
                NumTorre = NumTorre - 1
                frmdatos.Text1(1) = Val(frmdatos.Text1(1)) - 1
                If frmdatos.Grid1.row >= 2 Then
                    frmdatos.Grid1.row = frmdatos.Grid1.row - 1
                End If
            End If
        End With
    End If

GoTo fin

If Grid1.Rows = 10 Then
    With Grid1
        If Grid1.row > 1 Then
            If Grid1.TextMatrix(.row, 2) = "" Then
                MsgBox "No hay ninguna fila seleccionada para insertar.", vbExclamation, "Aerogenerador"
                Exit Sub
            End If
            If Grid1.TextMatrix(.row, 2) <> "" Then
                Grid1.RemoveItem .row
                Grid1.AddItem 10
                NumTorre = NumTorre + 1
                frmdatos.Text1(1) = Val(frmdatos.Text1(1)) + 1
            End If
        End If
    End With

End If
fin:

End Sub

```

Botón para insertar un tramo de torre

```

Private Sub cmdinsertar_Click()
    Dim row As Integer
    row = frmdatos.Grid1.row
    If Val(frmdatos.Text1(1)) = 0 Or frmdatos.Text1(1) = "" Then
        If row = 1 Then GoTo sigue
    End If

```

```
If frmdatos.Grid1.TextMatrix(row + 1, 1) = "" Then
    frmdatos.Grid1.row = frmdatos.Grid1.row + 1
End If
sigue:
ContadorTramos = 0
If frmdatos.Text2 <> "" Then
    CodigoTramos(1) = frmdatos.Text2
Else
    CodigoTramos(1) = frmdatos.Combo.Text
End If
CodigoTramos(ContadorTramos) = CodigoTramos(1)
Call InsertarDatosTorre
frmdatos.Text2 = ""
frmdatos.Combo.Text = frmdatos.Combo.List(0)
frmdatos.Text2.SetFocus

End Sub
```

Botón para realizar los cálculos de torre

```
Private Sub cmdcalculotorre_Click()
'Verificamos previamente que hayamos introducido todos los datos
For i = 2 To 13
    If Text1(i).Text = "" Then
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
        Text1(i).SetFocus
        Exit Sub
    End If
Next
'Verificamos que al menos se haya introducido un tramo de torre
If NumTorre = 0 Then
    MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"
    Exit Sub
End If
'Introducimos en la Matriz Dat todos los datos de la grúa
For i = 1 To 13
    Dat(i) = Val(Text1(i).Text)
Next
'Introducimos en la variable Tramo todos los datos de torre
For i = 1 To Dat(i)
    For j = 1 To 10
        Tramo(i, j) = frmdatos.Grid1.TextMatrix(i + 1, j + 2)
    Next
Next
Next
Call Calculo
```

Javier Vergara

frmcalculotorre.Show

frmdatos.Hide

fin:

End Sub

Botón para realizar los cálculos de la base más el pequeño estudio de cimentación

Private Sub cmdcalculotorre_Click()

'Verificamos previamente que hayamos introducido todos los datos

For i = 2 To 13

 If Text1(i).Text = "" Then

 MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"

 Text1(i).SetFocus

 Exit Sub

 End If

Next

'Verificamos que al menos se haya introducido un tramo de torre

If NumTorre = 0 Then

 MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"

 Exit Sub

End If

'Introducimos en la Matriz Dat todos los datos de la grúa

For i = 1 To 13

 Dat(i) = Val(Text1(i).Text)

Next

'Introducimos en la variable Tramo todos los datos de torre

For i = 1 To Dat(i)

 For j = 1 To 10

 Tramo(i, j) = frmdatos.Grid1.TextMatrix(i + 1, j + 2)

 Next

Next

Call Calculo

frmcalculobase.Show

frmdatos.Hide

fin:

End Sub

1.4.4 Botones de menú

Botón abrir archivo

```
Private Sub mnuabrir_Click()  
Dim Texto As String  
Dim posnombre As Integer  
'On Error GoTo final  
With CommonDialog1  
    .DialogTitle = "Abrir"  
    .FileName = RutaFichero  
    .Flags = cdIOFNFileMustExist  
    .ShowOpen  
    .CancelError = False  
  
    If Err.Number = 0 Then  
        If Len(.FileName) Then  
            RutaFichero = .FileName  
            posnombre = InStrRev(RutaFichero, ".")  
            Texto = Mid(RutaFichero, posnombre + 1)  
            cmdlimpiardatos_Click  
            cmdlimpiartramos_Click  
  
            'Comprobamos si es un archivo Excel o de texto  
            If Texto = "txt" Then  
                Call AbrirTexto  
            Else  
                Call AbrirExcel  
            End If  
        End If  
    End If  
End With  
final:  
  
End Sub
```

Botón abrir base de datos

```
Private Sub mnuabrirbase_Click()  
On Error GoTo error  
With CommonDialog1  
    .DialogTitle = "Abrir Base"  
    .FileName = RutaFichero  
    .Flags = cdIOFNFileMustExist  
    .ShowOpen  
    .CancelError = False  
    If Err.Number = 0 Then
```

```

    If Len(.FileName) Then
        RutaBaseDatos = .FileName
        RsTorre.Close
        DbBaseDatos.Close
        Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(RutaBaseDatos, True, True)
        NombreTabla = InputBox("Introduzca el nombre de la tabla por favor", "Aerogenerador")
        If NombreTabla = "" Then GoTo error
        Set RsTorre = DbBaseDatos.OpenRecordset(NombreTabla, dbOpenDynaset)
    End If
End If

End With

GoTo final

error:

MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical, "Aerogenerador"
Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal)
Set RsTorre = DbBaseDatos.OpenRecordset(TablaOriginal, dbOpenDynaset)
RutaBaseDatos = DirOriginal
NombreTabla = TablaOriginal

final:

Combo.Clear
RsTorre.MoveLast
RsTorre.MoveFirst
Combo.AddItem " "
For i = 1 To RsTorre.RecordCount
    Combo.AddItem RsTorre(0)
    RsTorre.MoveNext
Next

RsTorre.MoveFirst

frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos

End Sub

```

Botón para guardar los datos en archivo de texto

```

Private Sub mnuguardartexto_Click()
    'Verificamos que se hayan introducido todos los datos de la grúa
    For i = 2 To 13
        If Text1(i).Text = "" Then
            MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
            Text1(i).SetFocus
            'GoTo fin
            Exit Sub
        End If
    Next

```

Javier Vergara

```
        End If
    Next
    'Verificamos que se haya introducido por lo menos un tramo de torre
    If NumTorre = 0 Then
        MsgBox "Introduzca al menos un tramo de torre", vbOKOnly, "Aerogenerador"
        'GoTo fin
        Exit Sub
    End If

    Call DatosTexto
    frmhojaimprimir.Show
    fin:

End Sub
```

Botón para guardar los datos en archivo de Excel

```
Private Sub mnuguardarexcel_Click()
    'Verificamos que se hayan introducido todos los datos de la grúa
    For i = 2 To 13
        If Text1(i).Text = "" Then
            MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
            Text1(i).SetFocus
            'GoTo fin
            Exit Sub
        End If
    Next
    'Verificamos que se haya introducido por lo menos un tramo de torre
    If NumTorre = 0 Then
        MsgBox "Introduzca al menos un tramo de torre", vbOKOnly, "Aerogenerador"
        'GoTo fin
        Exit Sub
    End If

    Call DatosExcel
    fin:

End Sub
```

Botón para imprimir archivo

```
Private Sub mnuimprimir_Click()
    mnuguardartexto_Click
End Sub
```

Botón para salir del programa

```
Private Sub mnusalir_Click()
```

```
End  
End Sub
```

Botón para cortar

```
Private Sub mnuedicioncortar_Click()  
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
ext1(xi).SelText = ""  
  
End Sub
```

Botón para copiar

```
Private Sub mnuedicioncopiar_Click()  
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
mnuedicionpegar.Enabled = True  
  
End Sub
```

Botón para pegar

```
Private Sub mnuedicionpegar_Click()  
Text1(xi).SelText = Clipboard.GetText  
End Sub
```

Botones de Barra de Herramientas

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)  
Select Case Button.Key  
    Case "Open"  
        mnuabrir_Click  
    Case "Save"  
        'Verificamos que se hayan introducido todos los datos de la torre  
        For i = 2 To 13  
            If Text1(i).Text = "" Then  
                MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"  
                Text1(i).SetFocus  
                Exit Sub  
            End If  
        Next  
    End Case  
End Sub
```

```
'Verificamos que al menos se haya introducido un tramo de torre
If NumTorre = 0 Then
    MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"
    Exit Sub
End If
frmguardar.Show
Case "Print"
    mnuimprimir_Click
Case "Cut"
    mnuedicioncortar_Click
Case "Copy"
    mnuedicioncopiar_Click
Case "Paste"
    mnuedicionpegar_Click
'Case "Help"
' frm esquema.Show

End Select

End Sub
```


1.5 FORMULARIO CÁLCULO TORRE. FRMCAULOTORRE

1.5.1 Eventos de formulario

Cargar el formulario

For i = 1 To 2

With Grid1(i)

```
.TextArray(1) = "Tramo"
.TextArray(2) = "m"
.TextArray(3) = "Kg"
.TextArray(4) = "Cortante Kg"
.TextArray(5) = "Kg·m"
.TextArray(6) = "cm"
.TextArray(7) = "Montante"
.TextArray(8) = "Celosía Sup"
.TextArray(9) = "Celosía Inf"
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1150
.ColWidth(3) = 1150
.ColWidth(4) = 1300
.ColWidth(5) = 1400
.ColWidth(6) = 1400
.ColWidth(7) = 1400
.ColWidth(8) = 1400
.ColWidth(9) = 1400
.Rows = 16
```

End With

With Grid2(i)

```
.TextArray(1) = "Nº"
.TextArray(2) = "Altura"
.TextArray(3) = "Peso"
.TextArray(4) = "Esfuerzo"
.TextArray(5) = "Momento"
.TextArray(6) = "Flecha"
.TextArray(7) = "Tensión - Kg/cm²"
.TextArray(8) = "Tensión - Kg/cm²"
.TextArray(9) = "Tensión - Kg/cm²"
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1150
.ColWidth(3) = 1150
.ColWidth(4) = 1300
.ColWidth(5) = 1400
.ColWidth(6) = 1400
.ColWidth(7) = 1400
```

```

        .ColWidth(8) = 1400
        .ColWidth(9) = 1650
        .MergeRow(0) = True
    End With
Next

For i = 1 To 9
    Grid1(1).ColAlignment(i) = 4
    Grid2(1).ColAlignment(i) = 4
    Grid1(2).ColAlignment(i) = 4
    Grid2(2).ColAlignment(i) = 4
Next

If frmcalculotorre.Grid1(1).Rows < 14 Then    frmcalculotorre.Grid1(1).Rows = 14

If frmcalculotorre.Grid1(2).Rows < 14 Then frmcalculotorre.Grid1(2).Rows = 14

End Sub

```

1.5.2 Eventos de botones

Botón para guardar todos los datos y resultados en archivo de Excel

```

Private Sub cmdexcel_Click()
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then
    MousePointer = 11
    Call TodoResultadosExcel
    MousePointer = 0
End If
End Sub

```

Botón para guardar todos los datos y resultados en archivo de texto

```

Private Sub cmdtexto_Click()
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then
    MousePointer = 11
    Call DatosTexto
    Call TodoResultadosTexto
    MousePointer = 0
    frmhojaimprimir.Show
End If
End Sub

```

1.5.3 Botones de menú

Javier Vergara

Botón para guardar los resultados en archivo de texto

```
Private Sub mnuguardartexto_Click()  
MousePointer = 11  
Call DatosTexto  
Call ResultadosTorreTexto  
MousePointer = 0  
frmhojaimprimir.Show  
End Sub
```

Botón para guardar los resultados en archivo de Excel

```
Private Sub mnuguardarexcel_Click()  
Call ResultadosTorreExcel  
End Sub
```

Botón para imprimir los resultados

```
Private Sub mnuimprimir_Click()  
mnuguardartexto_Click  
End Sub
```

Botón para salir

```
Private Sub mnusalir_Click()  
End  
End Sub
```

Botón para cortar

```
Private Sub mnuedicioncortar_Click()  
Clipboard.Clear  
Clipboard.SetText Grid1(xi).Clip  
Grid1(xi).Clip = ""  
End Sub
```

Botón para copiar

```
Private Sub mnuedicioncopiar_Click()  
Clipboard.Clear  
Clipboard.SetText Grid1(xi).Clip  
mnuedicionpegar.Enabled = True  
End Sub
```

Botón de Barra de Herramientas

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Select Case Button.Key
```

```
    Case "Cut"
```

```
        mnuedicioncortar_Click
```

```
    Case "Copy"
```

```
        mnuedicioncopiar_Click
```

```
    Case "Paste"
```

```
        'mnuedicionpegar_Click
```

```
    Case "Print"
```

```
        mnuimprimir_Click
```

```
    Case "Save"
```

```
        frmguardar.Show
```

```
End Select
```

```
End Sub
```

1.6 FORMULARIO CÁLCULO DE BASE. FRMCALCULOBASE

1.6.1 Eventos de botones

Botón para guardar todos los datos y resultados en archivo de Excel

```
Private Sub cmdexcel_Click()
```

```
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then
```

```
    MousePointer = 11
```

```
    Call TodoResultadosExcel
```

```
    MousePointer = 0
```

```
End If
```

```
End Sub
```

Botón para guardar todos los datos y resultados en archivo de texto

```
Private Sub cmdtexto_Click()
```

```
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then
```

```
    MousePointer = 11
```

```
    Call DatosTexto
```

```
    Call TodoResultadosTexto
```

```
    MousePointer = 0
```

```
    frmhojaimprimir.Show
```

```
End If
```

```
End Sub
```

1.6.2 Botones de menú

Botón para guardar los resultados en archivo de texto

```
Private Sub mnuguardartexto_Click()  
MousePointer = 11  
Call DatosTexto  
Call ResultadosBaseTexto  
MousePointer = 0  
frmhojaimprimir.Show  
End Sub
```

Botón para guardar los resultados en archivo de Excel

```
Private Sub mnuguardarexcel_Click()  
Call ResultadosBaseExcel  
End Sub
```

Botón para imprimir los resultados

```
Private Sub mnuimprimir_Click()  
mnuguardartexto_Click  
End Sub
```

Botón para salir

```
Private Sub mnusalir_Click()  
End  
End Sub
```

Botón para cortar

```
Private Sub mnuedicioncortar_Click()  
Clipboard.Clear  
Clipboard.SetText Text1(xi).Text  
Text1(xi).Text = ""  
End Sub
```

Botón para copiar

Javier Vergara

```
Private Sub mnuedicioncopiar_Click()  
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
mnuedicionpegar.Enabled = True  
End Sub
```

Botón de Barra de Herramientas

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)  
Select Case Button.Key  
    Case "Cut"  
        mnuedicioncortar_Click  
    Case "Copy"  
        mnuedicioncopiar_Click  
    Case "Paste"  
        'mnuedicionpegar_Click  
    Case "Print"  
        mnuimprimir_Click  
    Case "Save"  
        frmguardar.Show  
End Select  
  
End Sub
```



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

***“Cálculo de aerogenerador en celosía mediante
software basado en Visual Basic”***

BIBLIOGRAFÍA

Javier Vergara Lecue

Tutores: Cesar Díaz de Cerio

Pamplona, 24 de Noviembre de 2010

Índice

	Página
1. LIBROS DE CONSULTA.....	3
2. NORMATIVA	3
3. PÁGINAS WEBS VISITADAS.....	3
4. PROGRAMAS INFORMÁTICOS UTILIZADOS.....	4

1 LIBROS DE CONSULTA

- HORMIGÓN ARMADO.
Autor: Pedro Jiménez Montoya
- RESISTENCIA DE MATERIALES
Autor: Luis Ortiz Berrocal
- TEORÍA DE ESTRUCTURAS
Autor: Jesús Zurita Gabasa
- BEGINNING VISUAL BASIC
Autor: Francisco Charte.
- Manual AIN. Visual Basic para Excel
Autor: Recursos humanos de fundación AIN
- Manual AIN. AUTOCAD 20100
Autor: Recursos humanos de fundación AIN

2 NORMATIVA

- EUROCÓDIGO Bases para el cálculo de estructuras
- EUROCÓDIGO 1 Parte 1.1 Acciones generales
- EUROCÓDIGO 1 Parte 1.4 Acciones en estructuras. Acciones Viento
- EUROCÓDIGO 3 Parte 1.1 Normas generales estructura metálica
- EUROCÓDIGO 3 Parte 3.1 Diseño de torres, mástiles y chimeneas

3 PAGINAS WEBS VISITADAS

- La web del programador.
- Microsoft.com/Visual-Studio

4 PROGRAMAS INFORMÁTICOS UTILIZADOS

- Autocad 2010.
- Visual Basic 6.0
- Microsoft Office



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

*“Cálculo de aerogenerador en celosía mediante
software basado en Visual Basic”*

ANEXO

Javier Vergara Lecue

Tutores: Cesar Díaz de Cerio

Pamplona, 24 de Noviembre de 2010

Índice

	Página
1.1 INTRODUCCIÓN.....	16
1.2 OBJETIVO DEL PROYECTO	19
PARTE 2: CONCEPTOS TEÓRICOS	21
CAPÍTULO 1: Esquema de los Cálculos.....	22
1.1 INTRODUCCIÓN.....	23
1.2 ANTECEDENTES DE LOS RRHH EN ESPAÑA	24
1.2.1 ETAPA ADMINISTRATIVA	24
1.2.2 ETAPA DE LAS RELACIONES HUMANAS.....	24
1.2.3 ETAPA DE LAS RELACIONES LABORALES	25
1.2.4 ETAPA DE LOS RECURSOS HUMANOS	26
1.2.5 ETAPA DE LA ORGANIZACIÓN ESTRATÉGICA	26
1.3 ESTRUCTURACIÓN Y FUNCIONES ACTUALES.....	28
CAPÍTULO 2: Prácticas relacionadas con la Dirección	31
2.1 DISEÑO DE LA ORGANIZACIÓN	32
2.1.1 ESTRUCTURA BUROCRÁTICA.....	32

1 INTRODUCCIÓN

En este Anexo se introducen todos los procedimientos que se han escrito en el programa. De manera que si alguna vez se desea adaptar el programa o verificar como trabaja, simplemente eligiendo el procedimiento desde este archivo lo obtendremos.

Se dividen los procedimientos por módulos, formularios, objetos...

Además de los procedimientos escritos tenemos las normas del EUROCÓDIGO utilizados.

2 MÓDULO CÁLCULOS

2.1 DECLARACIÓN DE VARIABLES

Option Explicit

Public Const Pi = 3.141592654

Public Tramo(80, 1 To 12) As Variant

Public Dat(1 To 30) As Double

Dim NInc(1 To 2) As Single

Dim AltInc(100) As Double

Dim AltViento(100) As Double

Dim MomentosCaso1(100, 4) As Double

Dim MomentosCaso2(100, 4) As Double

Dim MomentosSecunCaso1(100, 4) As Double

Dim MomentosSecunCaso2(100, 4) As Double

Dim Cr(100) As Double 'Coeficiente de Rugosidad

Dim Ct(100) As Double 'Coeficiente Topográfico

Dim Iv(100) As Double 'Intensidad de Turbulencia

Dim Ce(100) As Double 'Coeficiente de Exposición

Dim Cd As Double 'Coeficiente Dinámicos

Dim CF(80) As Double 'Coeficiente de Fuerza/Empuje

Dim CfMedio As Double 'Coeficiente de Fuerza medio

Dim g As Double 'Factor de pico

Dim Vm(100) As Double 'Velocidad Media Incremento

Dim Pre(100) As Double 'Presión del viento en cada incremento

Dim FVientoCaso1(100) As Double 'Fuerza del viento en cada incremento

Dim FVientoCaso2(100) As Double

Dim FVientoElementos(10) As Double 'Fuerza del viento en elementos, buje, caja, aspas...

Dim Peso(0 To 100) As Double 'Peso de cada incremento, teniendo en cuenta los pesos que tiene por encima

Dim EsfCortanteVientoCaso1(0 To 100) As Double 'Esfuerzo cortante en cada incremento

Dim EsfCortanteVientoCaso2(0 To 100) As Double 'Esfuerzo cortante en cada incremento

Dim Kt As Single 'Factor de terreno

Dim Zo As Single 'Longitud de rugosidad

Dim Zmin As Single 'Altura mínima

Dim Ancho As Single 'Anchura base

Dim Lh As Single 'Espesor Zapata

Dim e As Single 'Factor de categoría del terreno

Dim FlechasCaso1(100, 4) As Double

Dim FlechasCaso2(100, 4) As Double

Dim EsfCortante(2) As Double

Dim MomentoPeso As Double

Dim MomentoTotal(2) As Double

Dim MomentoSecundarioTotal(2) As Double

Javier Vergara

Dim Resultados(30) As Variant 'Guardaremos los resultados en esta variable

Dim Variables(30) As Double 'Guardaremos distintos valores momentos, esfuerzos cortantes...

2.2 FUNCIÓN CALCULO.

Sub Calculo()

'ENV 1991-2-4 Acciones del Viento Tabla 8.1 Categoría del Terreno 1

Kt = 0.17

Zo = 0.01

e = 0.13

Zmin = 2

'Anchura Base

Ancho = Tramo(Dat(1), 4)

Call Incremento

Call CoeficientesViento

Call FuerzaViento

Call CalculoMomentos_Esfuerzos

Call CalculoFlechas

Call CalculoMomentosSecundarios

Call CalculoEstructura

End Sub

Call CalculoBase

2.3 FUNCIÓN CÁLCULO DE INCREMENTOS.

Sub Incremento()

Dim i As Integer

Dim j As Integer

Dim k As Integer

'Con estas dos variables, sabemos cuántos incrementos tenemos en cada Tramo

NInc(1) = Int(100 / (Dat(1) + 2)) 'Nº de Incrementos por cada tramo de torre

NInc(2) = Int(Dat(1) + 2) * NInc(1) 'Nº de Incrementos Totales

k = 1

'División de la Torre

For i = 1 To Dat(1)

For j = 1 To NInc(1)

AltInc(k) = Tramo(i, 3) / NInc(1)

k = k + 1

Next

Next

End Sub

2.4 FUNCIÓN CÁLCULO DE COEFICIENTES DE VIENTO

Dim h As Single	'Altura total de la torre
Dim Zequ As Double	'Altura equivalente de la estructura
Dim IvZequ As Double	'Intensidad de turbulencia a la altura equivalente Zequ
Dim Qo As Double	'Respuesta de fondo
Dim Rx As Double	'Respuesta resonante
Dim v As Double	'Frecuencia esperada
Dim vo As Double	'Frecuencia esperada en HZ, de carga por ráfaga de viento en estructuras rígidas
Dim VmZequ As Double	'Velocidad media del viento a la altura equivalente Zequ
Dim LiZequ As Double	'Escala de longitud integral de turbulencia para la altura equivalente Zequ
Dim n1x As Double	'Frecuencia fundamental HZ, de carga por ráfaga de viento en estructuras rígidas
Dim Nx As Double	'Variable que se utiliza para calcular la función adimensional de la densidad espectral Rn
Dim Rn As Double	'Función adimensional de la densidad espectral de potencia
Dim Rh As Double	'Función de admitancia aerodinámica
Dim Rb As Double	'Función de admitancia aerodinámica
Dim Delta As Double	'Decremento logarítmico
Dim CrZequ As Double	'Coeficiente de rugosidad para la altura equivalente Zequ
Dim CtZequ As Double	'Coeficiente topográfico para la altura equivalente Zequ
Dim S As Double	'Variable que se utiliza para calcular la frecuencia esperada vo
Dim aux As Double	'Variable \square que se utiliza para calcular las funciones de admitancia Rh, Rb
Dim KDelta As Double	'Coeficiente de incidencia del viento
Dim K1 As Double	'Variable de coeficiente de incidencia
Dim K2 As Double	'Variable de coeficiente de incidencia
Dim CoefSolidez(80) As Double	'Coeficiente de Solidez
Dim C1 As Double	
Dim C2 As Double	
Dim Af As Double	'El área total proyectada, vista perpendicularmente, a la cara de los elementos con sección de lados planos en la cara.
Dim Ase As Double	'Sumatorio de todas las "áreas"
Dim Ac As Double	'El área total proyectada vista perpendicularmente a la cara, de los elementos de sección circular, que se encuentran en regímenes subcríticos
Coeficientes de empuje para secciones compuestas de elementos de sección de lados planos, circular subcrítica y circular supercrítica , respectivamente	
Dim Cfsoj As Double	
Dim Cfof As Double	
Dim Cfoc As Double	
Dim Cfocsup As Double	
For k = 1 To Dat(1)	
Af=Tramo(k,9)	
Ase = Af + Ac + Acsup	
K1 = 0.55 * Af / Ase + 0.8 * (Ac + Acsup) / Ase	

If $0 < \text{CoefSolidez}(k)$ And $\text{CoefSolidez}(k) < 0.2$ Then $K2 = 2$

If $0.8 < \text{CoefSolidez}(k)$ And $\text{CoefSolidez}(k) < 1$ Then $K2 = 2$

If $0.2 < \text{CoefSolidez}(k)$ And $\text{CoefSolidez}(k) < 0.5$ Then $K2 = \text{CoefSolidez}(k)$

If $0.5 < \text{CoefSolidez}(k)$ And $\text{CoefSolidez}(k) < 0.8$ Then $K2 = 1 - \text{CoefSolidez}(k)$

'Ojo incidencia del viento!!!!

$K\Delta = 1 + K1 * K2 * (\sin(45 * \pi / 180) * \cos(45 * \pi / 180)) ^ 2$

$C1 = 2.25$ 'Rectangular

$C2 = 1.5$ 'Rectangular

$C_{fof} = 1.76 * C1 * (1 - C2 * \text{CoefSolidez}(k) + \text{CoefSolidez}(k) ^ 2)$

$C_{foc} = C1 * (1 - C2 * \text{CoefSolidez}(k)) + (C1 + 0.875) + \text{CoefSolidez}(k) ^ 2$

$C_{soj} = C_{fof} * A_f / A_{se}$

$CF(k) = K\Delta * C_{soj}$

Next

$C_{fMedio} = 0$

For $i = 1$ To $Dat(1)$

$C_{fMedio} = C_{fMedio} + CF(i)$

Next

$C_{fMedio} = C_{fMedio} / Dat(1)$

$h = 0$

' Calcularemos la altura total de la torre

For $i = 1$ To $Dat(1)$

$h = h + \text{Tramo}(i, 3)$

Next

$Z_{equ} = 0.6 * h * Z_{min}$

Parámetros del viento y estructurales

$I_v(Z_{equ})$ La intensidad de turbulencia para la altura equivalente de la estructura

$Cr_{Zequ} = K_t * \log(Z_{equ} / Z_o)$ 'Coeficiente de rugosidad

Javier Vergara

$$CtZequ = 1$$

$$IvZequ = Kt / (CrZequ * CtZequ)$$

Factor de pico g. Debemos de calcular varios valores para poder llegar hasta el factor de pico g

$$LiZequ = 300 * (Zequ / 300) ^ e$$

$$VmZequ = Dat(20) * CrZequ * CtZequ$$

$$S = 0.46 * (Ancho + h) / LiZequ + 10.58 * ((Ancho * h) ^ (1 / 2)) / LiZequ$$

$$Qo = (1 / (1 + 0.9 * ((Ancho + h) / LiZequ) ^ 0.63)) ^ (1 / 2)$$

$$vo = VmZequ / LiZequ * 1 / (1.11 * S ^ 0.615)$$

Frecuencia Fundamental. ANEXO C.4 - Características Dinámicas

$$n1x = h / 46$$

$$Nx = n1x * LiZequ / VmZequ$$

$$Rn = 6.8 * Nx / (1 + 10.2 * Nx) ^ (5 / 3)$$

'Para calcular la masa equivalente fundamental por unidad de longitud

Call CalculoMasaEquivalente'Calculo del decremento logarítmico

$$Deltas = 0.05 \text{ 'Tabla C.8}$$

$$Deltaa = (1.25 * Ancho * CfMedio) / (2 * n1x * m1x) * VmZequ$$

$$Delta = Deltas + Deltaa$$

Calcularemos primero Rh

$$aux = 4.6 * Nx * h / LiZequ$$

If aux > 0 Then

$$Rh = 1 / aux - 1 / (2 * aux ^ 2) * (1 - \text{Exp}(-2 * aux))$$

End If

If aux = 0 Then

$$Rh = 1$$

End If

a continuación Rb

$$aux = 4.6 * Nx * Ancho / LiZequ$$

If aux > 0 Then

$$Rb = 1 / aux - 1 / (2 * aux ^ 2) * (1 - \text{Exp}(-2 * aux))$$

End If

Javier Vergara

If Int(aux) = 0 Then

$$Rb = 1$$

End If

$$Rx = (Pi^2 / (2 * Delta) * Rn * Rh * Rb)^{(1/2)}$$

$$v = ((vo^2 * Qo^2 + n1x^2 * Rx^2) / (Qo^2 + Rx^2))^{(1/2)}$$

$$g = (2 * Log(v * 600))^{(1/2)} + 0.6 / (2 * Log(v * 600))^{(1/2)}$$

'Coeficiente dinámico

$$Cd = (1 + 2 * g * IvZequ * (Qo^2 + Rx^2)^{(1/2)}) / (1 + 7 * IvZequ)$$

2.5 FUNCIÓN CÁLCULO DE LA FUERZA DEL VIENTO EN CADA INCREMENTO

```

Sub FuerzaViento()
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim h As Double 'Variable que se utiliza para calcular la altura respecto del suelo de cada incremento

h = 0

'Calcularemos la altura total de la torre
For i = 1 To Dat(1)
    h = h + Tramo(i, 3)
Next

'Calcularemos, el coeficiente de rugosidad, topográfico y de exposición de cada incremento.
'Así como la velocidad y presión media. Por último la fuerza que ejerce el viento en la torre para cada incremento

k = 1
For i = 1 To Dat(1)
    For j = 1 To NInc(1)
        AltViento(k) = h - AltInc(k) / 2
        'Coeficiente de rugosidad
        If AltViento(k) > Zo Then
            Cr(k) = Kt * Log(AltViento(k) / Zo)
        Else
            Cr(k) = Kt * Log(Zmin / Zo)
        End If
        'Coeficiente topográfico
        'Debo de preguntar
        Ct(k) = 1
        'Intensidad de turbulencia
        Iv(k) = Kt / (Cr(k) * Ct(k))
        'Coeficiente de exposición
        Ce(k) = Cr(k) ^ 2 * Ct(k) ^ 2 * (1 + 2 * g * Iv(k))
        'Velocidad media incremento' Faltarían por introducir
        Vm(k) = Dat(9) * Cr(k) * Ct(k)
        'Presión en cada incremento
        Pre(k) = 1.25 * 0.5 * Vm(k) ^ 2
        'Fuerza del viento en cada incremento
        FVientoCaso1(k) = Pre(k) * Ce(k) * Cd * CF(i) * Tramo(i, 9) / NInc(1)

        h = h - AltInc(k)
        k = k + 1
    Next
Next
  
```

End Sub

2.6 FUNCIÓN CÁLCULO DE MOMENTOS Y ESFUERZOS EN CADA INCREMENTO.

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim h As Double 'Variable que se utiliza para calcular la altura respecto del suelo de cada incremento

h = 0

For i = 1 To NInc(2)

AltViento(i) = 0

Next

'Calcularemos la altura total de la torre

For i = 1 To Dat(1)

h = h + Tramo(i, 3)

Next

'Calculamos la fuerza que el viento ejerce sobre la superficie de la góndola

'Coeficiente de rugosidad

$Cr(0) = Kt * \text{Log}((h + \text{Dat}(7) / 2) / Zo)$

'Coeficiente topográfico

$Ct(0) = 1$

'Intensidad de turbulencia

$Iv(0) = Kt / (Cr(0) * Ct(0))$

'Coeficiente de exposición

$Ce(0) = Cr(0) ^ 2 * Ct(0) ^ 2 * (1 + 2 * g * Iv(0))$

'Velocidad media incremento'

$Vm(0) = \text{Dat}(9) * Cr(0) * Ct(0)$

'Presión en cada incremento

$Pre(0) = 1.25 * 0.5 * Vm(0) ^ 2$

'Fuerza del viento

$FVientoElementos(1) = Pre(0) * Ce(0) * CF(1) * Cd * \text{Dat}(6)$

'Calculamos la fuerza del viento sobre las aspas

$$FVientoElementos(2) = Pre(0) * Pi * (Dat(8) / 2) ^ 2$$

'Calcularemos, el coeficiente de rugosidad, topográfico y de exposición de cada incremento.

'Así como la velocidad y presión media. Por último la fuerza que ejerce el viento en la torre para cada incremento

k = 1

For i = 1 To Dat(1)

For j = 1 To NInc(1)

$$AltViento(k) = h - AltInc(k) / 2$$

'Coeficiente de rugosidad

If AltViento(k) > Zmin Then

$$Cr(k) = Kt * Log(AltViento(k) / Zo)$$

Else

$$Cr(k) = Kt * Log(Zmin / Zo)$$

End If

'Coeficiente topográfico

$$Ct(k) = 1$$

'Intensidad de turbulencia

$$Iv(k) = Kt / (Cr(k) * Ct(k))$$

'Coeficiente de exposición

$$Ce(k) = Cr(k) ^ 2 * Ct(k) ^ 2 * (1 + 2 * g * Iv(k))$$

'Velocidad media incremento'

$$Vm(k) = Dat(9) * Cr(k) * Ct(k)$$

'Presión en cada incremento

$$Pre(k) = 1.25 * 0.5 * Vm(k) ^ 2$$

'Fuerza del viento en cada incremento

$$FVientoCaso1(k) = Pre(k) * Ce(k) * Cd * CF(i) * Tramo(i, 9) / (NInc(1))$$

$$FVientoCaso2(k) = FVientoCaso1(k) * 1.2$$

$$h = h - AltInc(k)$$

$$k = k + 1$$

Next

Next

End Sub

2.7 CÁLCULO DE LOS MOMENTOS FLECTORES, FUERZAS CORTANTES Y PESOS

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim l As Integer

Dim Dist(100) As Double

Dim DistBucle As Double

Dim Aux As Double

'Distancia respecto de la base de la parte superior del punto de incidencia del viento en la góndola y aspas

Dist(0) = Dat(7) / 2

'Momento Causado por el peso de la góndola y las aspas ya que están "descentrados"

MomentoPeso = (Dat(2) * Dat(4) + Dat(3) * Dat(5)) * 10

k = 0

For i = 1 To Dat(1)

For j = 1 To NInc(1)

k = k + 1

Dist(k) = Dist(k - 1) + AltInc(k)

DistBucle = AltInc(k) / 2

'CASO 1

'Momento causado por viento en la góndola

MomentosCaso1(k, 1) = FVientoElementos(1) * Dist(k)

'Momento causado por viento en aspas

MomentosCaso1(k, 2) = FVientoElementos(2) * Dist(k)

'Momento causado por el momento del peso cuasado por góndola y aspas

MomentosCaso1(k, 3) = MomentoPeso

'Momento casuado por viento en torre

$\text{MomentosCaso1}(k, 4) = \text{MomentosCaso1}(k, 4) + \text{FVientoCaso1}(k) * \text{DistBucle}$

'CASO 2

'Momento causado por viento en la góndola

$\text{MomentosCaso2}(k, 1) = \text{FVientoElementos}(1) * \text{Dist}(k)$

'Momento causado por viento en aspas

$\text{MomentosCaso2}(k, 2) = \text{FVientoElementos}(2) * \text{Dist}(k)$

'Momento causado por el momento del peso cuasado por góndola y aspas

$\text{MomentosCaso2}(k, 3) = \text{MomentoPeso}$

'Momento casuado por viento en torre

$\text{MomentosCaso2}(k, 4) = \text{MomentosCaso2}(k, 4) + \text{FVientoCaso2}(k) * \text{DistBucle}$

'Con este bucle sumaremos a cada "incremento", el momento causado por el viento que este por encima de cada "incremento"

For l = k + 1 To NInc(2)

$\text{DistBucle} = \text{DistBucle} + \text{AltInc}(l)$

$\text{MomentosCaso1}(l, 4) = \text{MomentosCaso1}(l, 4) + \text{FVientoCaso1}(k) * \text{DistBucle}$

$\text{MomentosCaso2}(l, 4) = \text{MomentosCaso2}(l, 4) + \text{FVientoCaso2}(k) * \text{DistBucle}$

Next

'Cálculo del esfuerzo cortante de cada incremento

$\text{EsfCortanteVientoCaso1}(k) = \text{EsfCortanteVientoCaso1}(k - 1) + \text{FVientoCaso1}(k)$

$\text{EsfCortanteVientoCaso2}(k) = \text{EsfCortanteVientoCaso2}(k - 1) + \text{FVientoCaso2}(k)$

'AUX = MomentosCaso1(96, 4)

Next

Next

End Sub

2.7 FUNCIÓN CÁLCULO DE FLECHAS.

Sub CalculoFlechas()

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim l As Integer

Dim Factor(100) As Double 'En esta variable guardaremos el valor de EIz de cada incremento, con el cambio de unidades

Dim DistEje As Double

Dim AlturaInc As Double

Dim Dist As Double

'Cálculo de las flechas producidas por el viento sobre la torre.

'Aplicaremos el segundo teorema de Mohr para hallar las flechas.

'Flecha A-B=Integral(Mz/EIz)x dx

'Calcularemos el momento de inercia de cada incremento.

k = 1

For i = 1 To Dat(1)

For j = 1 To NInc(1)

Factor(k) = Tramo(i, 6) * Tramo(i, 4) ^ 2 * 21000000

k = k + 1

Next

Next

k = 0

'Aplicación de Mohr

For j = 1 To 4

```
For i = 1 To NInc(2) 'Desde el primer hasta el último "incremento"

    Dist = 0

    For k = i To NInc(2)

        Dist = Dist + (AltInc(k) / 2)

        FlechasCaso1(i, j) = FlechasCaso1(i, j) + MomentosCaso1(k, j) * Dist * AltInc(k) / Factor(k) 'Momento Esfuerzos*Dist*Alt/F

        FlechasCaso2(i, j) = FlechasCaso2(i, j) + MomentosCaso2(k, j) * Dist * AltInc(k) / Factor(k) 'Momento Esfuerzos*Dist*Alt/F

        Dist = Dist + (AltInc(k) / 2)

    Next

Next

Next

Next

'For j = 1 To 4

'    For i = 1 To NInc(2)

'        FlechasCaso1(i, j) = FlechasCaso1(i, j) + MomentosCaso1(k, j) * AltViento(k) * AltInc(k) / Factor(k)

'    Next

'Next

End Sub
```

2.8 FUNCIÓN CÁLCULO DE MOMENTOS SECUNDARIOS

Sub CalculoMomentosSecundarios()

Dim i As Integer

Dim j As Integer

Dim k As Integer

'Cálculo de los momentos secundarios

'Una vez que hemos calculado las flechas, calcularemos los momentos secundarios que hay en cada incremento

'por la excentricidad de los pesos, derivada por las deformaciones

For k = 1 To 4 '(Nº de "Momentos" distintos que tenemos en la base, por cada una de las flechas de la anterior lista)

For i = 1 To NInc(2)

MomentosSecunCaso1(i, k) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso1(1, k) - FlechasCaso1(i, k))

MomentosSecunCaso2(i, k) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso2(1, k) - FlechasCaso2(i, k))

'Con este bucle sumaremos a cada "incremento", el momento causado por el viento que esté por encima de cada "incremento"

For j = 1 To i

MomentosSecunCaso1(i, k) = MomentosSecunCaso1(i, k) + Peso(j) * Dat(10) * (FlechasCaso1(j, k) - FlechasCaso1(i, k))

MomentosSecunCaso2(i, k) = MomentosSecunCaso2(i, k) + Peso(j) * Dat(10) * (FlechasCaso2(j, k) - FlechasCaso2(i, k))

Next

Next

Next

MomentoSecundarioTotal(1) = MomentosSecunCaso1(NInc(2), 1) + MomentosSecunCaso1(NInc(2), 2) +

MomentosSecunCaso1(NInc(2), 3) + MomentosSecunCaso1(NInc(2), 4)

MomentoSecundarioTotal(2) = MomentosSecunCaso2(NInc(2), 1) + MomentosSecunCaso2(NInc(2), 2) +

MomentosSecunCaso2(NInc(2), 3) + MomentosSecunCaso2(NInc(2), 4)

End Sub

2.9 FUNCIÓN CÁLCULO DE ESTRUCTURA

Sub CalculoEstructura()

Dim i As Integer

Dim ii As Integer

Dim j As Integer

Dim ki As Integer

Dim l As Integer

Dim EsfUnitMont(6) As Double

Dim EsfUnitDS(4) As Double

Dim EsfUnitDI(4) As Double

Dim MOM2Caso1(4) As Double

Dim MOM2Caso2(4) As Double

Javier Vergara

Dim T(9, 4) As Double

Dim aux(3) As Single

frmcalculotorre.Grid1(1).Rows = Dat(1) + 2

frmcalculotorre.Grid1(2).Rows = Dat(1) + 2

For i = 1 To Dat(1)

j = i * NInc(1)

'Puesta a cero de las Variables

For l = 1 To 6

EsfUnitMont(l) = 0

MOM2Caso1(l) = 0

MOM2Caso2(l) = 0

Next

For l = 1 To 4

EsfUnitDI(l) = 0

EsfUnitDS(l) = 0

Next

For l = 1 To 9

T(l, 1) = 0

T(l, 2) = 0

Next

Call Esfuerzo_Unitario(i, EsfUnitMont, EsfUnitDS, EsfUnitDI)

Call MomentosSecundarios_Unitario(j, MOM2Caso1, MOM2Caso2)

ki = i * NInc(1)

'CÁLCULO CASO 1-DIRECCIÓN HORIZONTAL

aux(1) = MOM2Caso1(1) + MOM2Caso1(2) + MOM2Caso1(3) + MOM2Caso1(4) 'Momentos Secundarios

aux(1) = MomentosSecunCaso1(ki, 1) + MomentosSecunCaso1(ki, 2) + MomentosSecunCaso1(ki, 3) + MomentosSecunCaso1(ki, 4)

T(1, 1) = i

T(2, 1) = 0 'Altura

T(3, 1) = (Dat(2) + Dat(3) + Peso(ki)) * Dat(10) 'Peso - Para los dos casos el mismo

T(4, 1) = FVientoElementos(1) + FVientoElementos(2) + EsfCortanteVientoCaso1(ki) 'Fuerza Cortante

T(5, 1) = MomentosCaso1(ki, 1) + MomentosCaso1(ki, 3) + MomentosCaso1(ki, 5) + MomentosCaso1(ki, 7) + aux(1)

T(6, 1) = FlechasCaso1(ki, 1) + FlechasCaso1(ki, 2) + FlechasCaso1(ki, 3) + FlechasCaso1(ki, 4)

T(7, 1) = T(5, 1) * EsfUnitMont(1) + T(3, 1) * EsfUnitMont(3) 'Tensión Montante

T(8, 1) = T(4, 1) * EsfUnitDS(1) 'Tensión Dia Sup

T(9, 1) = T(4, 1) * EsfUnitDI(1) 'Tensión Dia Inf

'CÁLCULO CASO 2-DIRECCIÓN DIAGONAL

aux(2) = MOM2Caso2(1) + MOM2Caso2(2) + MOM2Caso2(3) + MOM2Caso2(4) 'Momentos Secundarios

aux(2) = MomentosSecunCaso2(ki, 1) + MomentosSecunCaso2(ki, 2) + MomentosSecunCaso2(ki, 3) + MomentosSecunCaso2(ki, 4)

T(1, 2) = i

T(2, 2) = 0 'Altura

T(3, 2) = (Dat(2) + Dat(3) + Peso(ki)) * Dat(10) 'Peso - Para los dos casos el mismo

$T(4, 2) = F_{VientoElementos(1)} + F_{VientoElementos(2)} + EsfCortanteVientoCaso1(ki)$ 'Fuerza Cortante
 $T(5, 2) = MomentosCaso2(ki, 1) + MomentosCaso2(ki, 3) + MomentosCaso2(ki, 5) + MomentosCaso2(ki, 7) + aux(2)$
 $T(6, 2) = FlechasCaso2(ki, 1) + FlechasCaso2(ki, 2) + FlechasCaso2(ki, 3) + FlechasCaso2(ki, 4)$
 $T(7, 2) = T(5, 2) * EsfUnitMont(2) + T(3, 2) * EsfUnitMont(3)$ 'Tensión Montante
 $T(8, 2) = T(4, 2) * EsfUnitDS(2)$ 'Tensión Dia Sup
 $T(9, 2) = T(4, 2) * EsfUnitDI(2)$ 'Tensión Dia Inf

2.10 FUNCIÓN CÁLCULO DE ESFUERZOS UNITARIOS.

Sub Esfuerzo_Unitario(i, EsfUnitMont, EsfUnitDS, EsfUnitDI)

Dim Alfa As Double

'Ángulos de las Diagonales

Alfa = Atn(2 * Tramo(i, 3) / (Tramo(i, 2) * Tramo(i, 4)))

'Esfuerzos en el montante por flexión pura en dirección transversal a la sección. Caso 1

EsfUnitMont(1) = 1 / (2 * Tramo(i, 4) * Tramo(i, 6))

'Esfuerzos en la diagonal superior e inferior por esfuerzos cortante en dirección transversal a la sección. Caso 1

EsfUnitDS(1) = 1 / (2 * Tramo(i, 7))

EsfUnitDI(1) = 1 / (4 * Cos(Alfa) * Tramo(i, 8))

'Esfuerzos en el montante por flexión pura en dirección diagonal a la sección. Caso 2

EsfUnitMont(2) = 1 / (Sqr(2) * Tramo(i, 4) * Tramo(i, 6))

'Esfuerzos en la diagonal superior e inferior por esfuerzos cortante en dirección diagonal a la sección. Caso 2

EsfUnitDS(2) = 1 / (2 * Sqr(2) * Tramo(i, 7))

EsfUnitDI(2) = 1 / (4 * Sqr(2) * Cos(Alfa) * Tramo(i, 8))

'Esfuerzos en el montante debido al peso propio de la estructura, teniendo en cuenta del peso de buje/caja y aspas.

EsfUnitMont(3) = 1 / (4 * Tramo(i, 6))

End Sub

2.11 FUNCIÓN CÁLCULO DE MOMENTOS SECUNDARIOS UNITARIOS

Sub MomentosSecundarios_Unitario(j, MOM2Caso1, MOM2Caso2)

Dim l As Integer

Dim X As Integer

'MOMENTOS SECUNDARIOS

'Momentos Producidos por la Excentricidad del Peso Debida a las Distintas Flechas

'Al deformarse la torre, los pesos de esta quedan descentrados y producen unos momentos

'que aunque de escasa importancia también producen un esfuerzo

'Calcularemos el momento producido por el peso de la parte superior con excentricidad igual a la diferencia de flechas

'entre el primer incremento, que es el más alto, y el incremento más bajo del tramo

'y a continuación el momento que produce el peso de cada intervalo, con la diferencia de flechas de cada incremento
 'respecto a la base

'Los Momentos Secundarios FS solo son útiles en los casos MOMFS(1), MOMFS(3) y MOMFS(6)

```

For l = 1 To 4
  'If j < NinC(2) Then
    'MOM2Caso1(l) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso1(1, l) - FlechasCaso1(j + 1, l)) / 100
  ' Else
    MOM2Caso1(l) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso1(1, l) - FlechasCaso1(j, l)) / 100
    MOM2Caso2(l) = (Dat(2) + Dat(3)) * Dat(10) * (FlechasCaso2(1, l) - FlechasCaso2(j, l)) / 100
  'End If
  'For X = j - 1 To NinC(1) + 1 Step -1
  For X = 1 To j
    MOM2Caso1(l) = MOM2Caso1(l) + Peso(X) * Dat(10) * (FlechasCaso1(X, l) - FlechasCaso1(j, l))
    MOM2Caso2(l) = MOM2Caso2(l) + Peso(X) * Dat(10) * (FlechasCaso2(X, l) - FlechasCaso2(j, l))
  Next
Next
End
```

2.12 FUNCIÓN CÁLCULO DE BASE

```

Sub CalculoBase()
  Dim i As Integer
  Dim j As Integer
  Dim Peso As Double
  Dim Altura As Double
  Dim RPeso As Double
  Dim RMomentoVuelco(2) As Double

  'Puesta a 0 de las variables
  For i = 1 To 2
    EsfCortante(i) = 0
    MomentoTotal(i) = 0
  Next

  Peso = 0
  Altura = 0

  'Cálculo del peso y altura de la estructura
  For i = 1 To Dat(1)
    Altura = Altura + Tramo(i, 3)
    Peso = Peso + Tramo(i, 5)
  Next

  Peso = Peso + Dat(2) + Dat(3)
```

```

For i = 1 To NInc(2)
    EsfCortante(1) = EsfCortante(1) + FVientoCaso1(i)      'Dirección paralelo
    MomentoTotal(1) = MomentoTotal(1) + FVientoCaso1(i) * AltViento(i) 'Dirección paralelo
    EsfCortante(2) = EsfCortante(2) + FVientoCaso1(i)      'Dirección diagonal
    MomentoTotal(2) = MomentoTotal(2) + FVientoCaso1(i) * AltViento(i) 'Dirección diagonal
Next

MomentoTotal(1) = MomentoTotal(1) + MomentoSecundarioTotal(1) + MomentoPeso
MomentoTotal(2) = MomentoTotal(2) + MomentoSecundarioTotal(2) + MomentoPeso

'Cálculo de las fuerzas en los montantes debido al peso de la estructura
RPeso = Peso / 4

'Caso 1
'Cálculo de las fuerzas en los montantes debido al momento flector
RMomentoVuelco(1) = MomentoTotal(1) / (Ancho * 2)

'A continuación las sumaremos teniendo en cuenta si son de compresión o tracción
Resultados(1) = RPeso + RMomentoVuelco(1) 'Reacciones en 1 y 3
Resultados(2) = RPeso - RMomentoVuelco(1) 'Reacciones en 2 y 4

'Caso 2
'Cálculo de las fuerzas en los montantes debido al momento flector
RMomentoVuelco(2) = MomentoTotal(2) / (Ancho * Sqr(2))

'A continuación las sumaremos teniendo en cuenta si son de compresión o tracción
Resultados(3) = RPeso + RMomentoVuelco(2) 'Reacciones en 1
Resultados(4) = RPeso                      'Reacciones en 2 y 3
Resultados(5) = RPeso - RMomentoVuelco(2) 'Reacciones en 4

'Momentos de Vuelco
Resultados(6) = RMomentoVuelco(1) 'Dirección Horizontal. Caso 1
Resultados(7) = RMomentoVuelco(2) 'Dirección Diagonal. Caso 2

'Esfuerzos Cortantes
Resultados(8) = EsfCortante(1) 'Dirección Horizontal. Caso 1
Resultados(9) = EsfCortante(2) 'Dirección Diagonal. Caso 2

'Reacciones Máximas
Resultados(10) = Resultados(1) 'Dirección Horizontal. Caso 1
Resultados(11) = Resultados(3) 'Dirección Diagonal. Caso 2

'Peso Estructura
Resultados(12) = Peso

'Altura Estructura
Resultados(13) = Altura

'Asignaremos los resultados obtenidos a cada cuadro de texto en el formulario de resultados
'Reacciones Caso 1
  
```


frmcalculobase.Text1(1) = Resultados(1)
frmcalculobase.Text1(2) = Resultados(2)
frmcalculobase.Text1(3) = Resultados(1)
frmcalculobase.Text1(4) = Resultados(2)

'Reacciones Caso 2

frmcalculobase.Text1(5) = Resultados(3)
frmcalculobase.Text1(6) = Resultados(4)
frmcalculobase.Text1(7) = Resultados(4)
frmcalculobase.Text1(8) = Resultados(5)

'Momentos, Reacciones Horizontales, Reacciones Máximas

frmcalculobase.Text1(9) = Resultados(6)
frmcalculobase.Text1(10) = Resultados(7)
frmcalculobase.Text1(11) = Resultados(8)
frmcalculobase.Text1(12) = Resultados(9)
frmcalculobase.Text1(13) = Resultados(10)
frmcalculobase.Text1(14) = Resultados(11)

frmcalculobase.Label3(0) = Ancho

End Sub

2.13 CÁLCULO PEQUEÑO ESTUDIO DE CIMENTACIÓN

3 MÓDULO FUNCIONES

3.1 DECLARACIÓN DE VARIABLES

```
Option Explicit
Public RutaFichero As String
Public RutaBaseDatos As String
Public DirOriginal As String
Public NumTorre As Integer
Public DbBaseDatos As Database
Public RsTorre As Recordset
Public nFic As Long
Public lineatexto As String
Public CodigoTramos(80) As String
Public ContadorTramos As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer
```

3.2 FUNCIÓN ABRIR ARCHIVOS DE TEXTO

```
Sub AbrirTexto()
Dim lineasdatos(1000) As String
Dim lineastorre(1000) As String
Dim nlineas As Integer
Dim Pos(1000) As Integer
Dim PosTramos As Integer
Dim cont As Integer

nFic = FreeFile
nlineas = 0
PosTramos = 0

Open RutaFichero For Input As nFic
    While Not EOF(1)
        Line Input #nFic, lineatexto
        nlineas = nlineas + 1
    Wend
Close #nFic

Open RutaFichero For Input As nFic
'Repite el bucle hasta el final del archivo.
While Not EOF(1)
    For i = 1 To nlineas
        If i > 500 Then GoTo salimosbucle
```

```

'Lee el carácter en la variable.
Line Input #nFic, lineatexto
lineasdatos(i) = lineatexto
If PosTrafos <> 0 Then GoTo sigue
Pos(i) = InStrRev(lineasdatos(i), "=")
PosTrafos = InStr(lineasdatos(i), "Código")
If PosTrafos <> 0 Then
    PosTrafos = i
End If

sigue:

    Next
Wend

salimosbucle:

cont = 0

For i = 1 To nlineas
    If Not IsNumeric(Mid(lineasdatos(i), Pos(i) + 1, 7)) Then
        cont = cont + 1
    End If

    If IsNumeric(Mid(lineasdatos(i), Pos(i) + 1, 7)) Then
        lineasdatos(i - cont) = Mid(lineasdatos(i), Pos(i) + 1, 7)
    End If
Next

Close #nFic

For i = 1 To 13
    lineasdatos(i) = Replace(lineasdatos(i), ",", ".")
    frmdatos.Text1(i) = Replace(Val(lineasdatos(i)), ",", ".")
Next

For i = 1 To Val(frmdatos.Text1(1))
    lineastorre(i) = lineasdatos(PosTrafos + i + 2)
    lineastorre(i) = Left(lineastorre(i), 10)
    lineastorre(i) = Trim(lineastorre(i))
Next

'RutaBaseDatos = direccionbase
If RutaBaseDatos <> "" Then
    Call AbrirBase
Else
    MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical,
"Aerogenerador"
    RutaBaseDatos = DirOriginal
End If

```

```
'Insertamos los tramos de torre llamando al procedimiento InsertarDatosTorre
For k = 1 To Val(frmdatos.Text1(1))
    ContadorTramos = k
    CodigoTramos(ContadorTramos) = lineastorre(k)
    frmdatos.Grid1.row = k
    Call InsertarDatosTorre
Next

frmdatos.Text1(1) = Val(frmdatos.Text1(1)) / 2
For i = Val(frmdatos.Text1(1)) + 1 To frmdatos.Grid1.Rows - 1
    frmdatos.Grid1.TextMatrix(i, 1) = ""
Next

frmdatos.cmdcalculotorre.SetFocus

End Sub
```

3.3 FUNCIÓN ABRIR ARCHIVOS DE EXCEL

```
Sub AbrirExcel()
    Dim apli As Excel.Application
    Dim direccionbase As String
    Dim posbase As Integer
    'Creo el objeto
    Set apli = New Excel.Application
    'Abro el archivo seleccionado la hoja y celda deseada
    apli.Workbooks.Open RutaFichero
    apli.Sheets(1).Select
    apli.Range("b3").Select

    For i = 1 To 10
        frmdatos.Text1(i).Text = apli.ActiveCell.Offset(i, 0).Value
    Next

    apli.Range("b17").Select
    For i = 1 To 3
        frmdatos.Text1(i + 10).Text = apli.ActiveCell.Offset(i, 0).Value
    Next

    For i = 1 To 13
        frmdatos.Text1(i) = Replace(frmdatos.Text1(i), ",", ".")
    Next

    apli.Range("e5").Select
    For i = 1 To Val(frmdatos.Text1(1))
        CodigoTramos(i) = apli.ActiveCell.Offset(i, 0)
    Next

    direccionbase = apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 3)
```

```

posbase = InStrRev(direccionbase, "BASE DE DATOS: ")
direccionbase = Mid(direccionbase, posbase + 15)
RutaBaseDatos = direccionbase

If RutaBaseDatos <> "" Then
    'Call AbrirBase
Else
    MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical,
    "Aerogenerador"
    RutaBaseDatos = DirOriginal
End If

'Insertamos los tramos de torre llamando al procedimiento InsertarDatosTorre
For k = 1 To Val(frmdatos.Text1(1))
    ContadorTramos = k
    If k > frmdatos.Grid1.Rows - 2 Then
        MsgBox ("Se ha producido un error al insertar los datos de torre"), vbCritical, "Aerogenerador"
        GoTo final:
    End If
    CodigoTramos(ContadorTramos) = CodigoTramos(k)
    frmdatos.Grid1.row = k
    Call InsertarDatosTorre
Next

frmdatos.Text1(1) = Val(frmdatos.Text1(1)) / 2
For i = Val(frmdatos.Text1(1)) + 1 To frmdatos.Grid1.Rows - 1
    frmdatos.Grid1.TextMatrix(i, 1) = ""
Next

final:

apli.Workbooks.Close
apli.DisplayAlerts = False
apli.Quit

frmdatos.cmdcalculotorre.SetFocus
RutaBaseDatos = direccionbase

End Sub

```

3.4 FUNCIÓN PARA INSERTAR LOS DATOS DE TORRE DE LA BASE DE DATOS EN LA TABLA DEL FORMULARIO PRINCIPAL

```
Sub InsertarDatosTorre()  
Dim comilla As String  
Dim criterio As String  
Dim codigo As String  
Dim marca As String  
Dim row As Integer  
  
comilla = Chr(34)  
marca = RsTorre.Bookmark  
  
If frmdatos.Grid1.row = 0 Then GoTo final  
  
If CodigoTramos(ContadorTramos) = "" Then  
    MsgBox "Introduzca el Código del tramo de torre", vbExclamation, "Aerogenerador"  
    frmdatos.Grid1.row = frmdatos.Grid1.row - 1  
    GoTo final  
End If  
  
If NumTorre < 80 Then  
    codigo = CodigoTramos(ContadorTramos)  
    criterio = "Código like" & comilla & codigo & comilla  
  
    RsTorre.MoveFirst  
    RsTorre.FindFirst criterio  
    If RsTorre.NoMatch Then  
        MsgBox "No se encontró el tipo de tramo especificado. Vuelva a intentarlo.", vbInformation, "Aerogenerador"  
        RsTorre.Bookmark = marca  
        frmdatos.Grid1.row = frmdatos.Grid1.row - 1  
        GoTo final  
    Else  
        row = frmdatos.Grid1.row  
        If row = 0 Then row = 1  
        frmdatos.Grid1.AddItem "", row  
        For j = 1 To 11  
            frmdatos.Grid1.TextMatrix(row, j + 1) = RsTorre(j - 1)  
        Next  
        frmdatos.Grid1.TextMatrix(row, 1) = row  
        NumTorre = NumTorre + 1  
        frmdatos.Text1(1) = Val(frmdatos.Text1(1)) + 1  
        If NumTorre > 8 Then  
            frmdatos.Grid1.Rows = NumTorre + 4  
        End If  
    End If  
End If
```

```

    End If

Else

    MsgBox "No se pueden introducir más tramos de torre.", vbCritical, "Aerogenerador"
    frmdatos.Text2.Text = ""

End If

final:

RsTorre.MoveFirst
For i = 1 To Val(frmdatos.Text1(1))
    frmdatos.Grid1.TextMatrix(i, 1) = i
Next

End Sub

```

3.5 FUNCIÓN PARA ABRIR LA BASE DE DATOS CON LAS DIRECCIONES QUE OBTENEMOS DE LOS ARCHIVOS QUE ABRIMOS.

```

Sub AbrirBase()
    Dim NombreTabla As String
    Dim Pos As Integer
    On Error GoTo final1
    RsTorre.Close
    DbBaseDatos.Close

    Pos = InStrRev(RutaBaseDatos, ".mdb")

    If Pos <> 0 Then
        Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(RutaBaseDatos)
    Else
        Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(RutaBaseDatos + "\BaseDatos.mdb")
    End If

    NombreTabla = "Torre"
    'If NombreTabla = "" Then GoTo final1
    Set RsTorre = DbBaseDatos.OpenRecordset(NombreTabla, dbOpenDynaset)
    GoTo final

final1:

    MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical, "Aerogenerador"

    Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal + "\BaseDatos.mdb")
    Set RsTorre = DbBaseDatos.OpenRecordset("Torre", dbOpenDynaset)
    RutaBaseDatos = DirOriginal

final:

```

Javier Vergara

```
frmdatos.Combo.Clear
RsTorre.MoveLast
RsTorre.MoveFirst
frmdatos.Combo.AddItem " "
For i = 1 To RsTorre.RecordCount
    frmdatos.Combo.AddItem RsTorre(0)
    RsTorre.MoveNext
Next

RsTorre.MoveFirst

frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos

End Sub
```


4 MÓDULO GUARDARDATOS

4.1 DECLARACIÓN DE VARIABLES

```

Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim Texto As String
Dim Num(30) As String
Dim Numespacio As Integer
Dim apli As Excel.Application

```

4.2 FUNCIÓN PARA ESCRIBIR LOS DATOS DE ENTRADA EN LA CAJA DE TEXTO DEL FORMULARIO FRMHOJAIMPRIMIR

```

Sub DatosTexto()
Dim Label1(13) As String
For i = 1 To 9
    Num(i) = 0 & i
Next
For i = 10 To 26
    Num(i) = i
Next

```

```

Label1(1) = "    Número de Tramos....."
Label1(2) = "    Peso Góndola....."
Label1(3) = "    Peso Aspas....."
Label1(4) = "    Distancia CG Góndola....."
Label1(5) = "    Distancia hasta aspas....."
Label1(6) = "    Superficie Frontal Góndola....."
Label1(7) = "    Altura Góndola....."
Label1(8) = "    Diámetro aspas....."
Label1(9) = "    Viento de Referencia....."
Label1(10) = "    Coeficiente Peso Propio....."
Label1(11) = "    Densidad hormigón....."
Label1(12) = "    Tensión Admisible Terreno....."
Label1(13) = "    Altura Zapata....."

```

```
frmhojaimprimir.Text1 = vbCrLf & vbCrLf & Space(9) & "Fecha: " & Date & vbCrLf
```

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf & Space(40) & "DATOS TORRE"
```

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf
```

For i = 1 To 10

Texto = frmdatos.Text1(i)

Texto = Space(7 - Len(Texto)) + Texto

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Label1(i) & "Dat(" & Num(i) & ") =" & Texto & vbCrLf

If i = 1 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf

If i = 2 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg" & vbCrLf

If i = 3 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg" & vbCrLf

If i = 4 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf

If i = 5 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf

If i = 6 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m²" & vbCrLf

If i = 7 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf

If i = 8 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf

If i = 9 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m/s" & vbCrLf

If i = 10 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf

Next

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf & Space(37) & "DATOS CIMENTACIÓN"

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf

For i = 11 To 13

Texto = frmdatos.Text1(i)

Texto = Space(7 - Len(Texto)) + Texto

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Label1(i) & "Dat(" & Num(i) & ") =" & Texto & vbCrLf

If i = 11 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg/m" & vbCrLf

If i = 12 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Kg/m" & vbCrLf

If i = 13 Then frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " m" & vbCrLf

Next

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(38) & "TRAMOS DE TORRE" & vbCrLf

frmhojaimprimir.Text1	=	frmhojaimprimir.Text1	&	"
" & vbCrLf &				

vbCrLf

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " N°	Altura	Lg	Ancho	Peso	Sección cm²	SupViento	Tipo
Altura" & vbCrLf							

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Código Pasos m m Kg | Montante Celo.Sup Celo.Inf | m² Tramo  
Diagonal" & vbCrLf
```

```
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " " & vbCrLf &
```

```
For i = 1 To NumTorre
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(1)
```

```
    j = 2
```

```
    Texto = Left(frmdatos.Grid1.TextMatrix(i, j), 9)
```

```
    Numespacio = Int((9 - Len(Texto)) / 2)
```

```
    Texto = Space(Numespacio) + Texto + Space(Numespacio)
```

```
    Texto = Space(9 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(2)
```

```
    j = 3
```

```
    Texto = frmdatos.Grid1.TextMatrix(i, j)
```

```
    Texto = Space(3 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(2)
```

```
    j = 4
```

```
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)
```

```
    Texto = Space(6 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
```

```
    j = 5
```

```
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)
```

```
    Texto = Space(5 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
```

```
    j = 6
```

```
    Texto = frmdatos.Grid1.TextMatrix(i, j)
```

```
    Texto = Space(5 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)
```

```
    j = 7
```

```
    Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)
```

```
    Texto = Space(6 - Len(Texto)) + Texto
```

```
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
```

```

j = 8

Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)

Texto = Space(5 - Len(Texto)) + Texto

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)

j = 9

Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)

Texto = Space(5 - Len(Texto)) + Texto

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(8)

j = 10

Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)

Texto = Space(5 - Len(Texto)) + Texto

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(6)

j = 11

Texto = frmdatos.Grid1.TextMatrix(i, j)

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)

j = 12

Texto = FormatNumber(frmdatos.Grid1.TextMatrix(i, j), 2)

Texto = Space(5 - Len(Texto)) + Texto

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf

Next

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(38) & "BASE DE DATOS" & vbCrLf

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(10) & "Dirección: " & RutaBaseDatos & vbCrLf

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(10) & "Tabla: " & NombreTabla & vbCrLf

End Sub

```

4.3 FUNCIÓN PARA ESCRIBIR LOS RESULTADOS DE BASE Y DEL PEQUEÑO ESTUDIO DE CIMENTACIÓN EN LA CAJA DE TEXTO DEL FORMULARIO FRMHOJAIMPRIIR

Sub ResultadosBaseTexto()

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & vbCrLf & vbCrLf & Space(34) & "CÁLCULOS BASE" & vbCrLf & vbCrLf
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Space(35) & "Reacciones" & vbCrLf & vbCrLf & vbCrLf
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Space(20) & "1      2      3      4      " & vbCrLf
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & " _____"  
& vbCrLf & vbCrLf
```

```
For i = 1 To 2
```

```
    If i = 1 Then
```

```
        frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Space(4) & "Caso 1"
```

```
        For j = 1 To 4
```

```
            Texto = frmcalculobase.Text1(j)
```

```
            Texto = Space(7 - Len(Texto)) + Texto
```

```
            frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Texto & Space(6)
```

```
        Next
```

```
        frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & vbCrLf & vbCrLf
```

```
    End If
```

```
    If i = 2 Then
```

```
        frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Space(4) & "Caso 2"
```

```
        For j = 5 To 8
```

```
            Texto = frmcalculobase.Text1(j)
```

```
            Texto = Space(7 - Len(Texto)) + Texto
```

```
            frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Texto & Space(6)
```

```
        Next
```

```
    End If
```

```
Next
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & vbCrLf
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & " _____"  
& vbCrLf & vbCrLf
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Space(36) & "CASO 1" & vbCrLf & vbCrLf
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Space(20) & "Par de Vuelco....."
```

```
Texto = frmcalculobase.Text1(9)
```

```
Texto = Space(9 - Len(Texto)) + Texto
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Texto & " Kg·m" & vbCrLf
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Space(20) & "Reacción Horizontal....."
```

```
Texto = frmcalculobase.Text1(11)
```

```
Texto = Space(9 - Len(Texto)) + Texto
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Texto & " Kg" & vbCrLf
```

```
frmhojaimpriir.Text1 = frmhojaimpriir.Text1 & Space(20) & "Reacción Máxima....."
```

```
Texto = frmcalculobase.Text1(13)
```

```

Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg" & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(36) & "CASO 2" & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Par de Vuelco....."
Texto = frmcalculobase.Text1(10)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg·m" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Reacción Horizontal....."
Texto = frmcalculobase.Text1(12)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Reacción Máxima....."
Texto = frmcalculobase.Text1(14)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg" & vbCrLf & vbCrLf & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(30) & "CÁLCULOS CIMENTACIÓN" & vbCrLf & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Longitud Zapata....."
Texto = frmcalculobase.Text1(15)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " m" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Altura Zapata....."
Texto = frmcalculobase.Text1(16)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " m" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Tensión Zapata....."
Texto = frmcalculobase.Text1(17)
Texto = Space(9 - Len(Texto)) + Texto
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & " Kg/m²" & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(20) & "Distribución de Tensiones."
Texto = frmcalculobase.Text1(18)
Texto = Space(9 - Len(Texto)) + Texto
rmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & vbCrLf

End Sub

```

4.4 FUNCIÓN PARA ESCRIBIR LOS RESULTADOS DE TORRE EN LA CAJA DE TEXTO DEL FORMULARIO FRMHOJAIMPRIMIR

Sub ResultadosTorreTexto()

```

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(37) & "CÁLCULO DE TORRE"
For k = 1 To 2
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf & vbCrLf
    If k = 1 Then
        frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(32) & "CASO 1 - DIRECCIÓN HORIZONTAL" &
vbCrLf & vbCrLf
    End If

```

```

Else
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(33) & "CASO 2 - DIRECCIÓN DIAGONAL" &
vbCrLf & vbCrLf
End If

frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
_____ " & vbCrLf &
vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "  Nº  Altura  Peso  Esfuerzo  Momento  Flecha |    Tensión - Kg/cm²
| " & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & " Tramo    m    Kg  Cortante N  N·m    m |Montante  Celosía Sup Celosía
Inf| " & vbCrLf
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
_____ " & vbCrLf &
vbCrLf

For i = 1 To Dat(1)
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Space(3)
    j = 1
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
    Texto = Space(2 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
    j = 2
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
    Texto = Space(3 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
    j = 3
    Texto = Left(frmcalculotorre.Grid1(k).TextMatrix(i, j), 9)
    Numespacio = Int((9 - Len(Texto)) / 2)
    Texto = Space(Numespacio) + Texto + Space(Numespacio)
    Texto = Space(9 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(1)
    j = 4
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
    Texto = Space(7 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(2)
    j = 5
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
    Texto = Space(6 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
    j = 6
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
    Texto = Space(6 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(3)
    j = 7
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
    Texto = Space(6 - Len(Texto)) + Texto

```

```

    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(5)
    j = 8
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
    Texto = Space(5 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto & Space(4)
    j = 9
    Texto = frmcalculotorre.Grid1(k).TextMatrix(i, j)
    Texto = Space(5 - Len(Texto)) + Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & Texto
    frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & vbCrLf

Next
frmhojaimprimir.Text1 = frmhojaimprimir.Text1 & "
" & vbCrLf &
vbCrLf
Next

End Sub

```

4.5 FUNCIÓN PARA ESCRIBIR TANTO LOS DATOS COMO TODOS LOS RESULTADOS EN LA CAJA DE TEXTO DEL FORMULARIO FRMHOJAIMPRIMIR

```

Sub TodoResultadosTexto()
Call ResultadosTorreTexto
Call ResultadosBaseTexto
End Sub

```

4.6 FUNCIÓN PARA ESCRIBIR LOS DATOS DE ENTRADA EN UNA HOJA DE EXCEL

```

Sub DatosExcel()
Clipboard.Clear
Set apli = New Excel.Application
apli.Workbooks.Add
apli.Worksheets(1).Range("b1").Select
apli.WindowState = xlMaximized
apli.Visible = True
apli.ErrorCheckingOptions.BackgroundChecking = False
apli.Sheets("hoja1").Name = "Datos Torre"
apli.Columns(1).ColumnWidth = 29.29
apli.Columns(13).ColumnWidth = 14.57
apli.Range("a2").HorizontalAlignment = xlCenter
apli.Range("a2").Font.Bold = True

```



```

apli.Range("a2") = "DATOS TORRE"
apli.Range("a16").HorizontalAlignment = xlCenter
apli.Range("a16") = "DATOS CIMENTACIÓN"
apli.Range("a16").Font.Bold = True

apli.Range("a3").Select
For i = 1 To 10
    apli.ActiveCell.Offset(i, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i, 1).Value = Val(frmdatos.Text1(i))
Next

apli.Range("a17").Select
For i = 11 To 13
    apli.ActiveCell.Offset(i - 10, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i - 10, 1).Value = Val(frmdatos.Text1(i))
Next

apli.Columns("D:M").HorizontalAlignment = xlCenter
apli.Columns("D:M").NumberFormat = "@"
apli.Range("g2") = "TRAMOS DE TORRE"
apli.Range("g2").Font.Bold = True
apli.Range("g2", "i2").Merge
apli.Range("d4") = "Nº"
apli.Range("f4") = "Nº"
apli.Range("g4") = "Altura"
apli.Range("h4") = "Lg Ancho"
apli.Range("i4") = "Peso"
apli.Range("j4") = "Sección cm²"
apli.Range("j4", "l4").Merge
apli.Range("m4") = "Área Sup Viento"
apli.Range("d5") = "Tramo"
apli.Range("e5") = "Código"
apli.Range("f5") = "Pasos"
apli.Range("g5") = "m"
apli.Range("h5") = "m"
apli.Range("i5") = "Kg"
apli.Range("j5") = "Montante"
apli.Range("k5") = "Celosía Sup"
apli.Range("l5") = "Celosía Inf"
apli.Range("m5") = "Tramo m²"

apli.Range("d6").Select

Clipboard.Clear
frmdatos.Grid1.Col = 1
frmdatos.Grid1.Row = 1
frmdatos.Grid1.ColSel = 12
frmdatos.Grid1.RowSel = Val(frmdatos.Text1(1)) + 1

```

```
Clipboard.SetText frmdatos.Grid1.Clip
apli.ActiveSheet.Paste
frmdatos.Grid1.ColSel = 1
frmdatos.Grid1.RowSel = 1
Clipboard.Clear

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0) = "BASE DE DATOS"
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0), apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 1)).Merge
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0).HorizontalAlignment = xlLeft
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0) = "Dirección: " & RutaBaseDatos
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, 0) = "Tabla: " & NombreTabla

apli.Range("b1").Value = Date
apli.Range("a1").Select
et apli = Nothing

End Sub
```

4.7 FUNCIÓN PARA ESCRIBIR TANTO LOS DATOS COMO LOS RESULTADOS DE BASE Y DEL PEQUEÑO ESTUDIO DE CIMENTACIÓN EN UNA HOJA DE EXCEL

```
Sub ResultadosBaseExcel()
Clipboard.Clear
Set apli = New Excel.Application
apli.Workbooks.Add
apli.Worksheets(1).Range("b1").Select
apli.WindowState = xlMaximized
apli.Visible = True
apli.ErrorCheckingOptions.BackgroundChecking = False
apli.Sheets("hoja1").Name = "Datos Torre"
apli.Sheets("hoja2").Name = "Cálculos Base"
```

‘CÁLCULOS BASE/CIMENTACIÓN

```
apli.Sheets("Cálculos Base").Select
apli.Columns("A:I").HorizontalAlignment = xlCenter
apli.Columns(6).ColumnWidth = 17
apli.Columns(8).ColumnWidth = 17
apli.Range("c3") = "CÁLCULOS BASE"
apli.Range("c3").Font.Bold = True
apli.Range("c3", "d3").Merge
apli.Range("c6") = "Reacciones"
apli.Range("c6").Font.Bold = True
apli.Range("c6", "d6").Merge
```

```

apli.Range("f6") = "Caso 1"
apli.Range("f6").Font.Bold = True
apli.Range("f6", "g6").Merge
apli.Range("h6") = "Caso 2"
apli.Range("h6").Font.Bold = True
apli.Range("h6", "i6").Merge
apli.Range("a8") = "Caso 1"
apli.Range("a10") = "Caso 2"
apli.Range("f8") = "Par de Vuelco"
apli.Range("f9") = "Reacción Horizontal"
apli.Range("f10") = "Reacción Máxima"
apli.Range("h8") = "Par de Vuelco"
apli.Range("h9") = "Reacción Horizontal"
apli.Range("h10") = "Reacción Máxima"

apli.Range("b7") = 1
apli.Range("c7") = 2
apli.Range("d7") = 3
apli.Range("e7") = 4
apli.Range("c13") = "CÁLCULO CIMENTACIÓN"
apli.Range("c13").Font.Bold = True
apli.Range("c13", "d13").Merge
apli.Range("b15") = "Longitud Zapata"
apli.Range("b15", "c15").Merge
apli.Range("b16") = "Altura Zapata"
apli.Range("b16", "c16").Merge
apli.Range("b17") = "Tensión Zapata"
apli.Range("b17", "c17").Merge
apli.Range("b18") = "Distribución Tensiones"
apli.Range("b18", "c18").Merge

apli.Range("b8").Select
For i = 1 To 4
    apli.ActiveCell.Offset(0, i - 1) = Val(frmcalculobase.Text1(i))
    apli.ActiveCell.Offset(2, i - 1) = Val(frmcalculobase.Text1(i * 2))
Next
apli.Range("g8") = Val(frmcalculobase.Text1(9))
apli.Range("g9") = Val(frmcalculobase.Text1(11))
apli.Range("g10") = Val(frmcalculobase.Text1(13))
  
```

```

apli.Range("i8") = Val(frmcalculobase.Text1(10))
apli.Range("i9") = Val(frmcalculobase.Text1(12))
apli.Range("i10") = Val(frmcalculobase.Text1(14))
apli.Range("d15") = Val(frmcalculobase.Text1(15))
apli.Range("d16") = Val(frmcalculobase.Text1(16))
apli.Range("d17") = Val(frmcalculobase.Text1(17))
apli.Range("d18") = Val(frmcalculobase.Text1(1))
apli.Range("a1") = Date

```

```

apli.Range("a1").Select

```

'DATOS DE TORRE

```

apli.Sheets("Datos Torre").Select
apli.Columns(1).ColumnWidth = 29.29
apli.Columns(13).ColumnWidth = 14.57
apli.Range("a2").HorizontalAlignment = xlCenter
apli.Range("a2").Font.Bold = True
apli.Range("a2") = "DATOS TORRE"
apli.Range("a16").HorizontalAlignment = xlCenter
apli.Range("a16") = "DATOS CIMENTACIÓN"
apli.Range("a16").Font.Bold = True

```

```

apli.Range("a3").Select
For i = 1 To 10
    apli.ActiveCell.Offset(i, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i, 1).Value = Val(frmdatos.Text1(i))

```

Next

```

apli.Range("a17").Select
For i = 11 To 13
    apli.ActiveCell.Offset(i - 10, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i - 10, 1).Value = Val(frmdatos.Text1(i))

```

Next

```

apli.Columns("D:M").HorizontalAlignment = xlCenter
apli.Columns("D:M").NumberFormat = "@"
apli.Range("g2") = "TRAMOS DE TORRE"
apli.Range("g2").Font.Bold = True
apli.Range("g2", "i2").Merge
apli.Range("d4") = "Nº"
apli.Range("f4") = "Nº"
apli.Range("g4") = "Altura"
apli.Range("h4") = "Lg Ancho"
apli.Range("i4") = "Peso"
apli.Range("j4") = "Sección cm²"

```

```

apli.Range("j4", "l4").Merge
apli.Range("m4") = "Área Sup Viento"
apli.Range("d5") = "Tramo"
apli.Range("e5") = "Código"
apli.Range("f5") = "Pasos"
apli.Range("g5") = "m"
apli.Range("h5") = "m"
apli.Range("i5") = "Kg"
apli.Range("j5") = "Montante"
apli.Range("k5") = "Celosía Sup"
apli.Range("l5") = "Celosía Inf"
apli.Range("m5") = "Tramo m²"

apli.Range("d6").Select

Clipboard.Clear
frmdatos.Grid1.Col = 1
frmdatos.Grid1.Row = 1
frmdatos.Grid1.ColSel = 10
frmdatos.Grid1.RowSel = Val(frmdatos.Text1(1)) + 1
Clipboard.SetText frmdatos.Grid1.Clip
apli.ActiveSheet.Paste
frmdatos.Grid1.ColSel = 1
frmdatos.Grid1.RowSel = 1
Clipboard.Clear

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0) = "BASE DE DATOS"
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0), apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 1)).Merge
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0).HorizontalAlignment = xlLeft
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0) = "Dirección: " & RutaBaseDatos
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, 0) = "Tabla: " & NombreTabla

apli.Range("b1").Value = Date
apli.Range("a1").Select
Set apli = Nothing

End Sub

```

4.9 FUNCIÓN PARA ESCRIBIR TANTO LOS DATOS COMO LOS RESULTADOS DE TORRE EN UNA HOJA DE EXCEL

```

Sub ResultadosTorreExcel()
Clipboard.Clear
Set apli = New Excel.Application
apli.Workbooks.Add

```

```

apli.Worksheets(1).Range("b1").Select
apli.WindowState = xlMaximized
apli.Visible = True
apli.ErrorCheckingOptions.BackgroundChecking = False
apli.Sheets("hoja1").Name = "Datos Torre"
apli.Sheets("hoja2").Name = "Cálculos Torre"

```

'CÁLCULOS TORRE

```

Dim numfila As Integer
numfila = Dat(1)
apli.Sheets("Cálculos Torre").Select
apli.Columns("A:I").HorizontalAlignment = xlCenter
apli.Columns(4).ColumnWidth = 11.71
apli.Range("c3") = "CASO 1 - DIRECCIÓN HORIZONTAL"
apli.Range("c3").Font.Bold = True
apli.Range("c3", "f3").Merge
apli.Range("a5") = "Nº"
apli.Range("b5") = "Altura"
apli.Range("c5") = "Peso"
apli.Range("d5") = "Esfuerzo"
apli.Range("e5") = "Momento"
apli.Range("f5") = "Flecha"
apli.Range("g5") = "Tensión - Kg/cm²"
apli.Range("g5", "i5").Merge
apli.Range("a6") = "Tramo"
apli.Range("b6") = "m"
apli.Range("c6") = "Kg"
apli.Range("d6") = "Cortante - Kg"
apli.Range("e6") = "Kg·m"
apli.Range("f6") = "cm"
apli.Range("g6") = "Montante"
apli.Range("h6") = "Celosía Sup"
apli.Range("i6") = "Celosía Inf"

apli.Range("a6").Select
apli.ActiveCell.Offset(numfila + 2, 2) = "CASO 2 - DIRECCIÓN DIAGONAL"
apli.ActiveCell.Offset(numfila + 2, 2).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(numfila + 2, 2), apli.ActiveCell.Offset(numfila + 2, 5)).Merge
apli.ActiveCell.Offset(numfila + 4, 0) = "Nº"
apli.ActiveCell.Offset(numfila + 4, 1) = "Altura"
apli.ActiveCell.Offset(numfila + 4, 2) = "Peso"
apli.ActiveCell.Offset(numfila + 4, 3) = "Esfuerzo"
apli.ActiveCell.Offset(numfila + 4, 4) = "Momento"
apli.ActiveCell.Offset(numfila + 4, 5) = "Flecha"
apli.ActiveCell.Offset(numfila + 4, 6) = "Tensión - Kg/cm²"
apli.Range(apli.ActiveCell.Offset(numfila + 4, 6), apli.ActiveCell.Offset(numfila + 4, 8)).Merge

```

```

apli.ActiveCell.Offset(numfila + 5, 0) = "Tramo"
apli.ActiveCell.Offset(numfila + 5, 1) = "m"
apli.ActiveCell.Offset(numfila + 5, 2) = "Kg"
apli.ActiveCell.Offset(numfila + 5, 3) = "Cortante - Kg"
apli.ActiveCell.Offset(numfila + 5, 4) = "Kg·m"
apli.ActiveCell.Offset(numfila + 5, 5) = "cm"
apli.ActiveCell.Offset(numfila + 5, 6) = "Montante"
apli.ActiveCell.Offset(numfila + 5, 7) = "Celosía Sup"
apli.ActiveCell.Offset(numfila + 5, 8) = "Celosía Inf"
apli.Range("a7").Select
Clipboard.Clear
For j = 1 To 2
    Clipboard.Clear
    frmcalculotorre.Grid1(j).Col = 1
    frmcalculotorre.Grid1(j).row = 1
    frmcalculotorre.Grid1(j).ColSel = 9
    frmcalculotorre.Grid1(j).RowSel = numfila
    Clipboard.SetText frmcalculotorre.Grid1(j).Clip
    apli.ActiveSheet.Paste
    frmcalculotorre.Grid1(j).ColSel = 1
    frmcalculotorre.Grid1(j).RowSel = 1
    Clipboard.Clear
    apli.ActiveCell.Offset(numfila + 5, 0).Select
Next

apli.Range("a1") = Date
apli.Range("a1").Select

'DATOS DE TORRE

apli.Sheets("Datos Torre").Select
apli.Columns(1).ColumnWidth = 29.29
apli.Columns(13).ColumnWidth = 14.57
apli.Range("a2").HorizontalAlignment = xlCenter
apli.Range("a2").Font.Bold = True
apli.Range("a2") = "DATOS TORRE"
apli.Range("a16").HorizontalAlignment = xlCenter
apli.Range("a16") = "DATOS CIMENTACIÓN"
apli.Range("a16").Font.Bold = True

apli.Range("a3").Select
For i = 1 To 10
    apli.ActiveCell.Offset(i, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i, 1).Value = Val(frmdatos.Text1(i))
Next

apli.Range("a17").Select
For i = 11 To 13

```

```
apli.ActiveCell.Offset(i - 10, 0) = frmdatos.Label1(i)
apli.ActiveCell.Offset(i - 10, 1).Value = Val(frmdatos.Text1(i))
```

Next

```
apli.Columns("D:M").HorizontalAlignment = xlCenter
apli.Columns("D:M").NumberFormat = "@"
apli.Range("g2") = "TRAMOS DE TORRE"
apli.Range("g2").Font.Bold = True
apli.Range("g2", "i2").Merge
apli.Range("d4") = "Nº"
apli.Range("f4") = "Nº"
apli.Range("g4") = "Altura"
apli.Range("h4") = "Lg Ancho"
apli.Range("i4") = "Peso"
apli.Range("j4") = "Sección cm²"
apli.Range("j4", "l4").Merge
apli.Range("m4") = "Área Sup Viento"
apli.Range("d5") = "Tramo"
apli.Range("e5") = "Código"
apli.Range("f5") = "Pasos"
apli.Range("g5") = "m"
apli.Range("h5") = "m"
apli.Range("i5") = "Kg"
apli.Range("j5") = "Montante"
apli.Range("k5") = "Celosía Sup"
apli.Range("l5") = "Celosía Inf"
apli.Range("m5") = "Tramo m²"

apli.Range("d6").Select
Clipboard.Clear
frmdatos.Grid1.Col = 1
frmdatos.Grid1.Row = 1
frmdatos.Grid1.ColSel = 10
frmdatos.Grid1.RowSel = Val(frmdatos.Text1(1)) + 1
Clipboard.SetText frmdatos.Grid1.Clip
apli.ActiveSheet.Paste
frmdatos.Grid1.ColSel = 1
frmdatos.Grid1.RowSel = 1
Clipboard.Clear

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0) = "BASE DE DATOS"

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0), apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 1)).Merge
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0).HorizontalAlignment = xlLeft
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0) = "Dirección: " & RutaBaseDatos
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, 0) = "Tabla: " & NombreTabla
```



```
apli.Range("b1").Value = Date
apli.Range("a1").Select
Set apli = Nothing
```

```
End Sub
```

4.10 FUNCIÓN PARA ESCRIBIR TANTO LOS DATOS COMO TODOS LOS RESULTADOS EN UNA HOJA DE EXCEL

```
Sub TodoResultadosExcel()
Clipboard.Clear
Set apli = New Excel.Application
apli.Workbooks.Add
apli.Worksheets(1).Range("b1").Select
apli.WindowState = xlMaximized
apli.Visible = True
apli.ErrorCheckingOptions.BackgroundChecking = False
apli.Sheets("hoja1").Name = "Datos Torre"
apli.Sheets("hoja2").Name = "Cálculos Torre"
apli.Sheets("hoja3").Name = "Cálculos Base"
```

```
'CÁLCULOS BASE/CIMENTACIÓN
```

```
apli.Sheets("Cálculos Base").Select
apli.Columns("A:I").HorizontalAlignment = xlCenter
apli.Columns(6).ColumnWidth = 17
apli.Columns(8).ColumnWidth = 17
apli.Range("c3") = "CÁLCULOS BASE"
apli.Range("c3").Font.Bold = True
apli.Range("c3", "d3").Merge
apli.Range("c6") = "Reacciones"
apli.Range("c6").Font.Bold = True
apli.Range("c6", "d6").Merge
apli.Range("f6") = "Caso 1"
apli.Range("f6").Font.Bold = True
apli.Range("f6", "g6").Merge
apli.Range("h6") = "Caso 2"
apli.Range("h6").Font.Bold = True
apli.Range("h6", "i6").Merge
apli.Range("a8") = "Caso 1"
apli.Range("a10") = "Caso 2"
apli.Range("f8") = "Par de Vuelco"
apli.Range("f9") = "Reacción Horizontal"
apli.Range("f10") = "Reacción Máxima"
apli.Range("h8") = "Par de Vuelco"
apli.Range("h9") = "Reacción Horizontal"
```

```

apli.Range("h10") = "Reacción Máxima"
apli.Range("b7") = 1
apli.Range("c7") = 2
apli.Range("d7") = 3
apli.Range("e7") = 4
apli.Range("c13") = "CÁLCULO CIMENTACIÓN"
apli.Range("c13").Font.Bold = True
apli.Range("c13", "d13").Merge
apli.Range("b15") = "Longitud Zapata"
apli.Range("b15", "c15").Merge
apli.Range("b16") = "Altura Zapata"
apli.Range("b16", "c16").Merge
apli.Range("b17") = "Tensión Zapata"
apli.Range("b17", "c17").Merge
apli.Range("b18") = "Distribución Tensiones"
apli.Range("b18", "c18").Merge

apli.Range("b8").Select
For i = 1 To 4
    apli.ActiveCell.Offset(0, i - 1) = Val(frmcalculobase.Text1(i))
    apli.ActiveCell.Offset(2, i - 1) = Val(frmcalculobase.Text1(i * 2))
Next

apli.Range("g8") = Val(frmcalculobase.Text1(9))
apli.Range("g9") = Val(frmcalculobase.Text1(11))
apli.Range("g10") = Val(frmcalculobase.Text1(13))
apli.Range("i8") = Val(frmcalculobase.Text1(10))
apli.Range("i9") = Val(frmcalculobase.Text1(12))
apli.Range("i10") = Val(frmcalculobase.Text1(14))
apli.Range("d15") = Val(frmcalculobase.Text1(15))
apli.Range("d16") = Val(frmcalculobase.Text1(16))
apli.Range("d17") = Val(frmcalculobase.Text1(17))
apli.Range("d18") = Val(frmcalculobase.Text1(1))

apli.Range("a1") = Date
apli.Range("a1").Select

'CÁLCULOS TORRE

Dim numfila As Integer
numfila = Dat(1)
apli.Sheets("Cálculos Torre").Select
apli.Columns("A:I").HorizontalAlignment = xlCenter
apli.Columns(4).ColumnWidth = 11.71
apli.Range("c3") = "CASO 1 - DIRECCIÓN HORIZONTAL"
apli.Range("c3").Font.Bold = True
apli.Range("c3", "f3").Merge
  
```

```

apli.Range("a5") = "Nº"
apli.Range("b5") = "Altura"
apli.Range("c5") = "Peso"
apli.Range("d5") = "Esfuerzo"
apli.Range("e5") = "Momento"
apli.Range("f5") = "Flecha"
apli.Range("g5") = "Tensión - Kg/cm²"
apli.Range("g5", "i5").Merge
apli.Range("a6") = "Tramo"
apli.Range("b6") = "m"
apli.Range("c6") = "Kg"
apli.Range("d6") = "Cortante - Kg"
apli.Range("e6") = "Kg·m"
apli.Range("f6") = "cm"
apli.Range("g6") = "Montante"
apli.Range("h6") = "Celosía Sup"
apli.Range("i6") = "Celosía Inf"

apli.Range("a6").Select
apli.ActiveCell.Offset(numfila + 2, 2) = "CASO 2 - DIRECCIÓN DIAGONAL"
apli.ActiveCell.Offset(numfila + 2, 2).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(numfila + 2, 2), apli.ActiveCell.Offset(numfila + 2, 5)).Merge
apli.ActiveCell.Offset(numfila + 4, 0) = "Nº"
apli.ActiveCell.Offset(numfila + 4, 1) = "Altura"
apli.ActiveCell.Offset(numfila + 4, 2) = "Peso"
apli.ActiveCell.Offset(numfila + 4, 3) = "Esfuerzo"
apli.ActiveCell.Offset(numfila + 4, 4) = "Momento"
apli.ActiveCell.Offset(numfila + 4, 5) = "Flecha"
apli.ActiveCell.Offset(numfila + 4, 6) = "Tensión - Kg/cm²"
apli.Range(apli.ActiveCell.Offset(numfila + 4, 6), apli.ActiveCell.Offset(numfila + 4, 8)).Merge
apli.ActiveCell.Offset(numfila + 5, 0) = "Tramo"
apli.ActiveCell.Offset(numfila + 5, 1) = "m"
apli.ActiveCell.Offset(numfila + 5, 2) = "Kg"
apli.ActiveCell.Offset(numfila + 5, 3) = "Cortante - Kg"
apli.ActiveCell.Offset(numfila + 5, 4) = "Kg·m"
apli.ActiveCell.Offset(numfila + 5, 5) = "cm"
apli.ActiveCell.Offset(numfila + 5, 6) = "Montante"
apli.ActiveCell.Offset(numfila + 5, 7) = "Celosía Sup"
apli.ActiveCell.Offset(numfila + 5, 8) = "Celosía Inf"

apli.Range("a7").Select
Clipboard.Clear
For j = 1 To 2
    Clipboard.Clear
    frmcalculotorre.Grid1(j).Col = 1
    frmcalculotorre.Grid1(j).row = 1
    frmcalculotorre.Grid1(j).ColSel = 9
  
```

```

    frmcalculotorre.Grid1(j).RowSel = numfila
    Clipboard.SetText frmcalculotorre.Grid1(j).Clip
    apli.ActiveSheet.Paste
    frmcalculotorre.Grid1(j).ColSel = 1
    frmcalculotorre.Grid1(j).RowSel = 1
    Clipboard.Clear
    apli.ActiveCell.Offset(numfila + 5, 0).Select
Next

apli.Range("a1") = Date
apli.Range("a1").Select

'DATOS DE TORRE

apli.Sheets("Datos Torre").Select
apli.Columns(1).ColumnWidth = 29.29
apli.Columns(13).ColumnWidth = 14.57
apli.Range("a2").HorizontalAlignment = xlCenter
apli.Range("a2").Font.Bold = True
apli.Range("a2") = "DATOS TORRE"
apli.Range("a16").HorizontalAlignment = xlCenter
apli.Range("a16") = "DATOS CIMENTACIÓN"
apli.Range("a16").Font.Bold = True

apli.Range("a3").Select
For i = 1 To 10
    apli.ActiveCell.Offset(i, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i, 1).Value = Val(frmdatos.Text1(i))
Next

apli.Range("a17").Select
For i = 11 To 13
    apli.ActiveCell.Offset(i - 10, 0) = frmdatos.Label1(i)
    apli.ActiveCell.Offset(i - 10, 1).Value = Val(frmdatos.Text1(i))
Next

apli.Columns("D:M").HorizontalAlignment = xlCenter
apli.Columns("D:M").NumberFormat = "@"
apli.Range("g2") = "TRAMOS DE TORRE"
apli.Range("g2").Font.Bold = True
apli.Range("g2", "i2").Merge
apli.Range("d4") = "Nº"
apli.Range("f4") = "Nº"
apli.Range("g4") = "Altura"
apli.Range("h4") = "Lg Ancho"
apli.Range("i4") = "Peso"
apli.Range("j4") = "Sección cm²"
apli.Range("j4", "l4").Merge

```

```

apli.Range("m4") = "Área Sup Viento"
apli.Range("d5") = "Tramo"
apli.Range("e5") = "Código"
apli.Range("f5") = "Pasos"
apli.Range("g5") = "m"
apli.Range("h5") = "m"
apli.Range("i5") = "Kg"
apli.Range("j5") = "Montante"
apli.Range("k5") = "Celosía Sup"
apli.Range("l5") = "Celosía Inf"
apli.Range("m5") = "Tramo m²"

apli.Range("d6").Select
Clipboard.Clear
frmdatos.Grid1.Col = 1
frmdatos.Grid1.Row = 1
frmdatos.Grid1.ColSel = 10
frmdatos.Grid1.RowSel = Val(frmdatos.Text1(1)) + 1
Clipboard.SetText frmdatos.Grid1.Clip
apli.ActiveSheet.Paste
frmdatos.Grid1.ColSel = 1
frmdatos.Grid1.RowSel = 1
Clipboard.Clear

apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0) = "BASE DE DATOS"
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0).Font.Bold = True
apli.Range(apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 0), apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 1, 1)).Merge
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0).HorizontalAlignment = xlLeft
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 2, 0) = "Dirección: " & RutaBaseDatos
apli.ActiveCell.Offset(Val(frmdatos.Text1(1)) + 3, 0) = "Tabla: " & NombreTabla

apli.Range("b1").Value = Date
apli.Range("a1").Select
Set apli = Nothing

End Sub

```

4 FORMULARIO DE DATOS.

4.1 EVENTOS DE FORMULARIO

Cargar el formulario

Describimos las tablas

With Grid1

.TextArray(1) = "Tramo"

.TextArray(2) = "Código"

.TextArray(3) = "Pasos"

.TextArray(4) = "m"

.TextArray(5) = "m"

.TextArray(6) = "Kg"

.TextArray(7) = "Montante"

.TextArray(8) = "Celosía Sup"

.TextArray(9) = "Celosía Inf"

.TextArray(10) = "Tramo m²"

.TextArray(11) = "Tramo"

.TextArray(12) = "Diagonal"

.ColWidth(0) = 0

.ColWidth(1) = 800

.ColWidth(2) = 1100

.ColWidth(3) = 800

.ColWidth(4) = 900

.ColWidth(5) = 950

.ColWidth(6) = 950

.ColWidth(7) = 1000

.ColWidth(8) = 1350

.ColWidth(9) = 1350

.ColWidth(10) = 1700

End With

With Grid2

.RowHeight(0) = 450

.TextArray(1) = "Nº"

.TextArray(3) = "Nº"

.TextArray(4) = "Altura"

.TextArray(5) = "Lg Ancho"

.TextArray(6) = "Peso"

.TextArray(7) = "Sección cm²"

.TextArray(8) = "Sección cm²"

.TextArray(9) = "Sección cm²"

.TextArray(10) = "Área Sup Viento"

.TextArray(11) = "Tipo"

.TextArray(12) = "Altura "

.ColWidth(0) = 0

.ColWidth(1) = 800

.ColWidth(2) = 1100

.ColWidth(3) = 800

.ColWidth(4) = 900

.ColWidth(5) = 950

.ColWidth(6) = 950

.ColWidth(7) = 1000

.ColWidth(8) = 1350

.ColWidth(9) = 1350

.ColWidth(10) = 1700

Javier Vergara

```
.ColWidth(12) = 1330

.MergeRow(0) = True

End With

'Damos la orientación de texto deseada a las columnas

For i = 1 To 12

    Grid1.ColAlignment(i) = 4

    Grid2.ColAlignment(i) = 4

Next

'Abrimos el texto y obtenemos la dirección de la base de datos

nFic = FreeFile

Open "BaseDatos.cfg" For Input As nFic

    Line Input #nFic, lineatexto

    Line Input #nFic, lineatexto

    i = InStr(lineatexto, Chr(34))

    j = InStr(i + 1, lineatexto, Chr(34))

    RutaBaseDatos = Mid(lineatexto, i + 1, j - i - 1)

    DirOriginal = RutaBaseDatos

Close

DirOriginal = DirOriginal + "\BaseDatos"

RutaBaseDatos = DirOriginal

TablaOriginal = "Aerogenerador"

NombreTabla = TablaOriginal

'Abrimos la base de datos de los tramos de torre

Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal)

Set RsTorre = DbBaseDatos.OpenRecordset(TablaOriginal, dbOpenDynaset)

'Llevamos todos los códigos de tramo al combo

RsTorre.MoveLast
```


Javier Vergara

```
RsTorre.MoveFirst

Combo.AddItem " "

For i = 1 To RsTorre.RecordCount

    Combo.AddItem RsTorre(0)

    RsTorre.MoveNext

Next

RsTorre.MoveFirst

'Al índice de las cajas de texto le asignamos el valor 2

xi = 2

End Sub
```

Activar el formulario

```
Private Sub Form_Activate()
frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos
End Sub
```

Arrastrar algún archivo sobre el formulario

```
Private Sub Form_OLEDragDrop(Data As DataObject, Effect As Long, Button As Integer, Shift As Integer, X As Single, Y As Single)
'On Error GoTo Final

'Si no es un archivo lo que se arrastra, habrá error
If Data.GetFormat(vbCFFiles) = False Then
    MsgBox "No se ha arrastrado ningún archivo", vbInformation
    Exit Sub
End If

'Si se arrastra más de un archivo, habrá error
If Data.Files.Count > 1 Then
    MsgBox "No se puede arrastrar más de un archivo"
    Exit Sub
End If

cmdlimpiardatos_Click
cmdlimpiartramos_Click
```

Javier Vergara

```

Dim posnombre As Integer
Dim Texto As String
RutaFichero = Data.Files(1)
posnombre = InStrRev(RutaFichero, ".")
Texto = Mid(RutaFichero, posnombre + 1)

'Comprobamos si es un archivo Excel o de texto
If Texto = "txt" Then
    Call AbrirTexto
Else
    Call AbrirExcel
End If

final:

End Sub
  
```

Descargar el formulario

```

Private Sub Form_Unload(Cancel As Integer)
End
End Sub
  
```

4.2 EVENTOS EN CAJAS DE TEXTO

Al pulsar cualquier tecla sobre Text1

```

Private Sub Text1_KeyPress(Index As Integer, KeyAscii As Integer)
Dim X As Integer
'Sólo se pueden teclear en los textos número y "."
If KeyAscii <> Asc("9") Then
If KeyAscii <> Asc("8") Then
If KeyAscii <> Asc("7") Then
If KeyAscii <> Asc("6") Then
If KeyAscii <> Asc("5") Then
If KeyAscii <> Asc("4") Then
If KeyAscii <> Asc("3") Then
If KeyAscii <> Asc("2") Then
If KeyAscii <> Asc("1") Then
If KeyAscii <> Asc("0") Then
If KeyAscii <> 8 Then      'tecla suprimir
    X = 1
End If
End If
End If
End If
End If
  
```

```
End If
End If
End If
End If
End If
End If
If KeyAscii = Asc(".") Then
    X = 2
End If
If KeyAscii = Asc(",") Then
    X = 3
End If
If KeyAscii = vbKeyReturn Then
    X = 4
End If

If X = 1 Then
    KeyAscii = 0
    MsgBox "Introduzca sólo números", vbInformation
End If
If X = 2 Then
    KeyAscii = Asc(".")
End If
If X = 3 Then
    KeyAscii = Asc(".")
End If
If X = 4 Then
    xi = xi + 1
    If xi = 14 Then
        Combo.SetFocus
        xi = 2
        Exit Sub
    End If
    Text1(xi).SetFocus
End If

End Sub
```

Al hacer Click con el ratón sobre la caja de texto Text1

```
Private Sub Text1_Click(Index As Integer)
    xi = Index
End Sub
```

Al pulsar cualquier tecla sobre Text2

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    cmdinsertar_Click
End If

End Sub
```

4.3 EVENTOS DE BOTONES

Botón para bajar un tramo de torre

```
Private Sub cmdbajar_Click()
Dim tramos(12) As String
With Grid1
    If Grid1.TextMatrix(.row, 2) = "" Then
        MsgBox "Seleccione un tramo de torre para desplazarlo.", vbInformation, "Aerogenerador"
        Exit Sub
    End If
    If Grid1.row = NumTorre Then Exit Sub
    For j = 2 To 12
        tramos(j) = Grid1.TextMatrix(.row + 1, j)
        Grid1.TextMatrix(.row + 1, j) = Grid1.TextMatrix(.row, j)
        Grid1.TextMatrix(.row, j) = tramos(j)
    Next
    Grid1.row = Grid1.row + 1
    Grid1.TopRow = Grid1.row - 1
    Grid1.Col = 0
    Grid1.ColSel = 10
End With

End Sub
```

Botón para subir un tramo de torre

```
Private Sub cmdsubir_Click()  
Dim tramos(12) As String  
With Grid1  
    If Grid1.TextMatrix(.row, 2) = "" Then  
        MsgBox "Seleccione un tramo de torre para desplazarlo.", vbInformation, "Aerogenerador"  
        Exit Sub  
    End If  
    If Grid1.row = 1 Then Exit Sub 'GoTo final  
    For j = 2 To 12  
        tramos(j) = Grid1.TextMatrix(.row - 1, j)  
        Grid1.TextMatrix(.row - 1, j) = Grid1.TextMatrix(.row, j)  
        Grid1.TextMatrix(.row, j) = tramos(j)  
    Next  
    Grid1.row = Grid1.row - 1  
    If Grid1.row <> 1 Then  
        Grid1.TopRow = Grid1.row - 1  
    End If  
    Grid1.Col = 0  
    Grid1.ColSel = 10  
End With  
  
End Sub
```

Botón para eliminar un tramo de torre

```
Private Sub cmdeliminar_Click()  
If Grid1.Rows > 10 Then  
  
    With Grid1  
        If Grid1.TextMatrix(.row, 2) = "" Then  
            MsgBox "No hay ninguna fila seleccionada para eliminar.", vbExclamation, "Aerogenerador"  
            Exit Sub  
        Else  
            If frmdatos.Grid1.row = 0 Then frmdatos.Grid1.row = 1  
            .RemoveItem .row  
            NumTorre = NumTorre - 1  
            frmdatos.Text1(1) = Val(frmdatos.Text1(1)) - 1  
        End If  
    End With  
End Sub
```

```
If frmdatos.Grid1.row >= 2 Then

    frmdatos.Grid1.row = frmdatos.Grid1.row - 1

End If

End If

End With

End If

GoTo fin

If Grid1.Rows = 10 Then

    With Grid1

        If Grid1.row > 1 Then

            If Grid1.TextMatrix(.row, 2) = "" Then

                MsgBox "No hay ninguna fila seleccionada para insertar.", vbExclamation, "Aerogenerador"

                Exit Sub

            End If

            If Grid1.TextMatrix(.row, 2) <> "" Then

                Grid1.RemoveItem .row

                Grid1.AddItem 10

                NumTorre = NumTorre - 1

                frmdatos.Text1(1) = Val(frmdatos.Text1(1)) - 1

            End If

        End If

    End With

End If

fin:

For i = 1 To NumTorre

    frmdatos.Grid1.TextMatrix(i, 1) = i
```

Next
 End Sub

Botón para insertar un tramo de torre

```
Private Sub cmdinsertar_Click()
Dim row As Integer
row = frmdatos.Grid1.row
If Val(frmdatos.Text1(1)) = 0 Or frmdatos.Text1(1) = "" Then
    If row = 1 Then GoTo sigue
End If
If frmdatos.Grid1.TextMatrix(row + 1, 1) = "" Then
    frmdatos.Grid1.row = frmdatos.Grid1.row + 1
End If
sigue:
ContadorTramos = 0
If frmdatos.Text2 <> "" Then
    CodigoTramos(1) = frmdatos.Text2
Else
    CodigoTramos(1) = frmdatos.Combo.Text
End If
CodigoTramos(ContadorTramos) = CodigoTramos(1)
Call InsertarDatosTorre
frmdatos.Text2 = ""
frmdatos.Combo.Text = frmdatos.Combo.List(0)
frmdatos.Text2.SetFocus

End Sub
```

Botón para realizar los cálculos de torre

```
Private Sub cmdcalculotorre_Click()
'Verificamos previamente que hayamos introducido todos los datos
For i = 2 To 13
    If Text1(i).Text = "" Then
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
        Text1(i).SetFocus
        Exit Sub
    End If
Next
'Verificamos que al menos se haya introducido un tramo de torre
If NumTorre = 0 Then
    MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"
    Exit Sub
End If
```

```

End If
'Introducimos en la Matriz Dat todos los datos de la grúa
For i = 1 To 13
    Dat(i) = Val(Text1(i).Text)
Next
'Introducimos en la variable Tramo todos los datos de torre
For i = 1 To Dat(i)
    For j = 1 To 12
        Tramo(i, j) = frmdatos.Grid1.TextMatrix(i + 1, j + 2)
    Next
Next
Call Calculo
frmcalculotorre.Show
frmdatos.Hide

fin:

End Sub
  
```

Botón para realizar los cálculos de la base más el pequeño estudio de cimentación

```

Private Sub cmdcalculotorre_Click()
'Verificamos previamente que hayamos introducido todos los datos
For i = 2 To 13
    If Text1(i).Text = "" Then
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
        Text1(i).SetFocus
        Exit Sub
    End If
Next
'Verificamos que al menos se haya introducido un tramo de torre
If NumTorre = 0 Then
    MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"
    Exit Sub
End If
'Introducimos en la Matriz Dat todos los datos de la grúa
For i = 1 To 13
    Dat(i) = Val(Text1(i).Text)
Next
'Introducimos en la variable Tramo todos los datos de torre
For i = 1 To Dat(i)
    For j = 1 To 12
        Tramo(i, j) = frmdatos.Grid1.TextMatrix(i + 1, j + 2)
    Next
Next
Call Calculo
  
```



```
frmcalculobase.Show  
frmdatos.Hide  
fin:  
End Sub
```

4.4 BOTONES DE MENÚ

Botón abrir archivo

```
Private Sub mnuabrir_Click()  
Dim Texto As String  
Dim posnombre As Integer  
'On Error GoTo final  
With CommonDialog1  
    .DialogTitle = "Abrir"  
    .FileName = RutaFichero  
    .Flags = cdIOFNFileMustExist  
    .ShowOpen  
    .CancelError = False  
  
    If Err.Number = 0 Then  
        If Len(.FileName) Then  
            RutaFichero = .FileName  
            posnombre = InStrRev(RutaFichero, ".")  
            Texto = Mid(RutaFichero, posnombre + 1)  
            cmdlimpiardatos_Click  
            cmdlimpiartramos_Click  
  
            'Comprobamos si es un archivo Excel o de texto  
            If Texto = "txt" Then  
                Call AbrirTexto  
            Else  
                Call AbrirExcel  
            End If  
        End If  
    End With  
final:  
  
End Sub
```

Botón abrir base de datos

```

Private Sub mnuabrirbase_Click()
On Error GoTo error
With CommonDialog1
    .DialogTitle = "Abrir Base"
    .FileName = RutaFichero
    .Flags = cdIOFNFileMustExist
    .ShowOpen
    .CancelError = False
    If Err.Number = 0 Then
        If Len(.FileName) Then
            RutaBaseDatos = .FileName
            RsTorre.Close
            DbBaseDatos.Close
            Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(RutaBaseDatos, True, True)
            NombreTabla = InputBox("Introduzca el nombre de la tabla por favor", "Aerogenerador")
            If NombreTabla = "" Then GoTo error
            Set RsTorre = DbBaseDatos.OpenRecordset(NombreTabla, dbOpenDynaset)
        End If
    End If
End With

GoTo final

error:

MsgBox "Se ha producido un error al abrir la base de datos. Se utilizará la base de datos por defecto", vbCritical, "Aerogenerador"
Set DbBaseDatos = DBEngine.Workspaces(0).OpenDatabase(DirOriginal)
Set RsTorre = DbBaseDatos.OpenRecordset(TablaOriginal, dbOpenDynaset)
RutaBaseDatos = DirOriginal
NombreTabla = TablaOriginal

final:

Combo.Clear
RsTorre.MoveLast
RsTorre.MoveFirst
Combo.AddItem " "
For i = 1 To RsTorre.RecordCount
    Combo.AddItem RsTorre(0)
    RsTorre.MoveNext
Next

RsTorre.MoveFirst

frmdatos.Caption = "Aerogenerador\ Base Datos " & RutaBaseDatos
  
```

End Sub

Botón para guardar los datos en archivo de texto

```
Private Sub mnuguardartexto_Click()  
'Verificamos que se hayan introducido todos los datos de la grúa  
For i = 2 To 13  
    If Text1(i).Text = "" Then  
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"  
        Text1(i).SetFocus  
        'GoTo fin  
        Exit Sub  
    End If  
Next  
'Verificamos que se haya introducido por lo menos un tramo de torre  
If NumTorre = 0 Then  
    MsgBox "Introduzca al menos un tramo de torre", vbOKOnly, "Aerogenerador"  
    'GoTo fin  
    Exit Sub  
End If  
  
Call DatosTexto  
frmhojaimprimir.Show  
fin:  
  
End Sub
```

Botón para guardar los datos en archivo de Excel

```
Private Sub mnuguardarexcel_Click()  
'Verificamos que se hayan introducido todos los datos de la grúa  
For i = 2 To 13  
    If Text1(i).Text = "" Then  
        MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"  
        Text1(i).SetFocus  
        'GoTo fin  
        Exit Sub  
    End If  
Next  
'Verificamos que se haya introducido por lo menos un tramo de torre  
If NumTorre = 0 Then  
    MsgBox "Introduzca al menos un tramo de torre", vbOKOnly, "Aerogenerador"  
    'GoTo fin  
    Exit Sub  
End If  
  
Call DatosExcel  
fin:  
  
End Sub
```

Botón para imprimir archivo

```
Private Sub mnuimprimir_Click()  
mnuguardartexto_Click  
End Sub
```

Botón para salir del programa

```
Private Sub mnusalir_Click()  
End  
End Sub
```

Botón para cortar

```
Private Sub mnuedicioncortar_Click()  
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
ext1(xi).SelText = ""  
  
End Sub
```

Botón para copiar

```
Private Sub mnuedicioncopiar_Click()  
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
mnuedicionpegar.Enabled = True  
  
End Sub
```

Botón para pegar

```
Private Sub mnuedicionpegar_Click()  
Text1(xi).SelText = Clipboard.GetText  
End Sub
```

Botones de Barra de Herramientas

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Key
    Case "Open"
        mnuabrir_Click
    Case "Save"
        'Verificamos que se hayan introducido todos los datos de la torre
        For i = 2 To 13
            If Text1(i).Text = "" Then
                MsgBox "Falta algún valor por introducir. Vuelva a intentarlo.", vbInformation, "Aerogenerador"
                Text1(i).SetFocus
                Exit Sub
            End If
        Next
        'Verificamos que al menos se haya introducido un tramo de torre
        If NumTorre = 0 Then
            MsgBox "Introduzca al menos un tramo de torre", vbInformation, "Aerogenerador"
            Exit Sub
        End If
        frmguardar.Show
    Case "Print"
        mnuimprimir_Click
    Case "Cut"
        mnuedicioncortar_Click
    Case "Copy"
        mnuedicioncopiar_Click
    Case "Paste"
        mnuedicionpegar_Click
    'Case "Help"
    ' frm esquema.Show
End Select

End Sub
```

5 FORMULARIO CÁLCULO TORRE. FRMCALULOTORRE

5.1 EVENTOS DE FORMULARIO

Cargar el formulario

For i = 1 To 2

With Grid1(i)

```
.TextArray(1) = "Tramo"
.TextArray(2) = "m"
.TextArray(3) = "Kg"
.TextArray(4) = "Cortante N"
.TextArray(5) = "N·m"
.TextArray(6) = "m"
.TextArray(7) = "Montante"
.TextArray(8) = "Celosía Sup"
.TextArray(9) = "Celosía Inf"
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1150
.ColWidth(3) = 1150
.ColWidth(4) = 1300
.ColWidth(5) = 1400
.ColWidth(6) = 1400
.ColWidth(7) = 1400
.ColWidth(8) = 1400
.ColWidth(9) = 1400
.Rows = 16
```

End With

With Grid2(i)

```
.TextArray(1) = "Nº"
.TextArray(2) = "Altura"
.TextArray(3) = "Peso"
.TextArray(4) = "Esfuerzo"
.TextArray(5) = "Momento"
.TextArray(6) = "Flecha"
.TextArray(7) = "Tensión - Kg/cm²"
.TextArray(8) = "Tensión - Kg/cm²"
.TextArray(9) = "Tensión - Kg/cm²"
.ColWidth(0) = 0
.ColWidth(1) = 800
.ColWidth(2) = 1150
.ColWidth(3) = 1150
```

```

        .ColWidth(4) = 1300
        .ColWidth(5) = 1400
        .ColWidth(6) = 1400
        .ColWidth(7) = 1400
        .ColWidth(8) = 1400
        .ColWidth(9) = 1650
        .MergeRow(0) = True
    End With
Next

For i = 1 To 9
    Grid1(1).ColAlignment(i) = 4
    Grid2(1).ColAlignment(i) = 4
    Grid1(2).ColAlignment(i) = 4
    Grid2(2).ColAlignment(i) = 4
Next

If frmcalculotorre.Grid1(1).Rows < 14 Then    frmcalculotorre.Grid1(1).Rows = 14

If frmcalculotorre.Grid1(2).Rows < 14 Then frmcalculotorre.Grid1(2).Rows = 14

End Sub

```

5.2 EVENTOS DE BOTONES

Botón para guardar todos los datos y resultados en archivo de Excel

```

Private Sub cmdexcel_Click()
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then
    MousePointer = 11
    Call TodoResultadosExcel
    MousePointer = 0
End If
End Sub

```

Botón para guardar todos los datos y resultados en archivo de texto

```

Private Sub cmdtexto_Click()
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then
    MousePointer = 11
    Call DatosTexto
    Call TodoResultadosTexto
    MousePointer = 0
    frmhojaimprimir.Show
End If
End Sub

```

```
End If  
End Sub
```

5.3 BOTONES DE MENÚ

Botón para guardar los resultados en archivo de texto

```
Private Sub mnuguardartexto_Click()  
MousePointer = 11  
Call DatosTexto  
Call ResultadosTorreTexto  
MousePointer = 0  
frmhojaimprimir.Show  
End Sub
```

Botón para guardar los resultados en archivo de Excel

```
Private Sub mnuguardarexcel_Click()  
Call ResultadosTorreExcel  
End Sub
```

Botón para imprimir los resultados

```
Private Sub mnuimprimir_Click()  
mnuguardartexto_Click  
End Sub
```

Botón para salir

```
Private Sub mnusalir_Click()  
End  
End Sub
```

Botón para cortar

```
Private Sub mnuedicioncortar_Click()  
Clipboard.Clear  
Clipboard.SetText Grid1(xi).Clip  
Grid1(xi).Clip = ""  
End Sub
```


Javier Vergara

Botón para copiar

```
Private Sub mnuedicioncopiar_Click()  
Clipboard.Clear  
Clipboard.SetText Grid1(xi).Clip  
mnuedicionpegar.Enabled = True  
End Sub
```

Botón de Barra de Herramientas

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)  
Select Case Button.Key  
    Case "Cut"  
        mnuedicioncortar_Click  
    Case "Copy"  
        mnuedicioncopiar_Click  
    Case "Paste"  
        'mnuedicionpegar_Click  
    Case "Print"  
        mnuimprimir_Click  
    Case "Save"  
        frmguardar.Show  
End Select  
  
End Sub
```

6 FORMULARIO CÁLCULO DE BASE. FRMCALCULOBASE

6.1 EVENTOS DE BOTONES

Botón para guardar todos los datos y resultados en archivo de Excel

```
Private Sub cmdexcel_Click()  
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then  
    MousePointer = 11  
    Call TodoResultadosExcel  
    MousePointer = 0  
End If  
End Sub
```

Botón para guardar todos los datos y resultados en archivo de texto

```
Private Sub cmdtexto_Click()  
If MsgBox("Se tardará unos segundos en realizar la siguiente tarea...", vbOKCancel, "Aerogenerador") = vbOK Then  
    MousePointer = 11  
    Call DatosTexto  
    Call TodoResultadosTexto  
    MousePointer = 0  
    frmhojaimprimir.Show  
End If  
End Sub
```

6.2 BOTONES DE MENÚ

Botón para guardar los resultados en archivo de texto

```
Private Sub mnuguardartexto_Click()  
MousePointer = 11  
Call DatosTexto  
Call ResultadosBaseTexto  
MousePointer = 0  
frmhojaimprimir.Show  
End Sub
```

Botón para guardar los resultados en archivo de Excel

Javier Vergara

```
Private Sub mnguardarexcel_Click()  
Call ResultadosBaseExcel  
End Sub
```

Botón para imprimir los resultados

```
Private Sub mnuimprimir_Click()  
mnguardartexto_Click  
End Sub
```

Botón para salir

```
Private Sub mnusalir_Click()  
End  
End Sub
```

Botón para cortar

```
Private Sub mnuedicioncortar_Click()  
Clipboard.Clear  
Clipboard.SetText Text1(xi).Text  
Text1(xi).Text = ""  
End Sub
```

Botón para copiar

```
Private Sub mnuedicioncopiar_Click()  
Clipboard.Clear  
Clipboard.SetText Text1(xi).SelText  
mnuedicionpegar.Enabled = True  
End Sub
```

Botón de Barra de Herramientas

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)  
Select Case Button.Key  
Case "Cut"  
mnuedicioncortar_Click  
Case "Copy"  
mnuedicioncopiar_Click  
Case "Paste"  
mnuedicionpegar_Click  
End Select
```

Javier Vergara

```
Case "Print"  
    mnuimprimir_Click  
Case "Save"  
    frmguardar.Show
```

```
End Select
```

```
End Sub
```

7 FORMULARIO HOJAGUARDAR. FRMHOJAIMPRIMIR

Botón de Barra de Herramientas

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Select Case Button.Key
```

```
    Case "Cut"
```

```
        mnuedicioncortar_Click
```

```
    Case "Copy"
```

```
        mnuedicioncopiar_Click
```

```
    Case "Paste"
```

```
        'mnuedicionpegar_Click
```

```
    Case "Print"
```

```
        mnuimprimir_Click
```

```
    Case "Save"
```

```
        frmguardar.Show
```

```
End Select
```

```
End Sub
```

Botón guardar

```
Private Sub mnuguardar_Click()
```

```
    'Dim RutaFichero As String
```

```
    Dim nFic As Long
```

```
    On Error GoTo final
```

```
    With CommonDialog1
```

```
        .DialogTitle = "Guardar configuración"
```

```
        .FileName = RutaFichero
```

```
        .Flags = cdIOFNOverwritePrompt
```

```
        .Filter = "Documentos de texto (*.txt)*.txt|Todos los archivos*.*"
```

```
        .ShowSave
```

```
        .CancelError = False
```

```
    If Len(.FileName) Then
```

```
        RutaFichero = .FileName
```

```
        nFic = FreeFile
```

```
    Open RutaFichero For Output As nFic
```

Javier Vergara

Print #nFic, Text1.Text

Close nFic

End If

End With

final:

End Sub

English Version

**Eurocode 3 - Design of steel structures - Part 3-1: Towers,
masts and chimneys - Towers and masts**

Eurocode 3 - Calcul des structures en acier - Partie 3-1:
Tours, mâts et cheminées - Pylônes et mâts haubannés

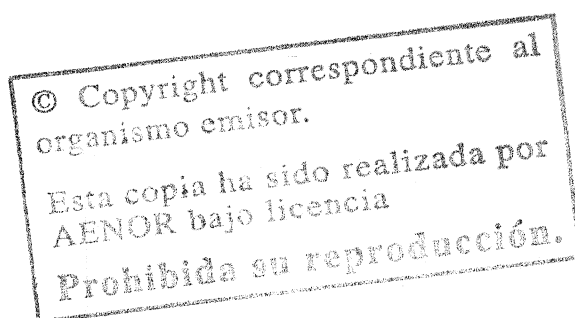
Eurocode 3 - Bemessung und Konstruktion von
Stahlbauten - Teil 3-1: Türme, Maste und Schornsteine -
Türme und Maste

This European Standard was approved by CEN on 9 January 2006.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the Central Secretariat has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

Contents

1	General	9
1.1	Scope	9
1.2	Normative references	9
1.3	Assumptions	10
1.4	Distinction between principles and application rules	10
1.5	Terms and definitions	10
1.6	Symbols	11
1.7	Convention for cross section axes	12
2	Basis of design	13
2.1	Requirements	13
2.2	Principles of limit state design	14
2.3	Actions and environmental influences	14
2.4	Ultimate limit state verifications	15
2.5	Design assisted by testing	15
2.6	Durability	15
3	Materials	16
3.1	Structural steel	16
3.2	Connections	16
3.3	Guys and fittings	16
4	Durability	16
4.1	Allowance for corrosion	16
4.2	Guys	16
5	Structural analysis	17
5.1	Modelling for determining action effects	17
5.2	Modelling of connections	17
6	Ultimate limit states	18
6.1	General	18
6.2	Resistance of cross sections	18
6.3	Resistance of members	18
6.4	Connections	20
6.5	Special connections for masts	21
7	Serviceability limit states	23
7.1	Basis	23
7.2	Deflections and rotations	23
7.3	Vibrations	23
8	Design assisted by testing	24
9	Fatigue	24
9.1	General	24
9.2	Fatigue loading	24
9.3	Fatigue resistance	25
9.4	Safety assessment	25
9.5	Partial factors for fatigue	25
9.6	Fatigue of guys	25
Annex A [normative] – Reliability differentiation and partial factors for actions		26
A.1	Reliability differentiation for masts and towers	26

A.2	Partial factors for actions.....	26
Annex B [informative]	– Modelling of meteorological actions	27
B.1	General	27
B.2	Wind force.....	28
B.3	Response of lattice towers.....	40
B.4	Response of guyed masts	45
Annex C [informative]	– Ice loading and combinations of ice with wind.....	53
C.1	General	53
C.2	Ice loading	53
C.3	Ice weight.....	54
C.4	Wind and ice	54
C.5	Asymmetric ice load	54
C.6	Combinations of ice and wind.....	55
Annex D [normative]	– Guys, dampers, insulators, ancillaries and other items	56
D.1	Guys	56
D.2	Dampers	56
D.3	Insulators	57
D.4	Ancillaries and other items.....	57
Annex E [informative]	– Guy rupture	59
E.1	Introduction.....	59
E.2	Simplified analytical model	59
E.3	Conservative procedure.....	60
E.4	Analysis after a guy rupture	61
Annex F [informative]	– Execution.....	62
F.1	General	62
F.2	Bolted connections	62
F.3	Welded connections	62
F.4	Tolerances	62
F.5	Prestretching of guys.....	63
Annex G [informative]	– Buckling of components of masts and towers.....	64
G.1	Buckling resistance of compression members	64
G.2	Effective slenderness factor k	64
Annex H [informative]	– Buckling length and slenderness of members	70
H.1	General	70
H.2	Leg members.....	70
H.3	Bracing members	71
H.4	Secondary bracing members	78
H.5	Shell structures.....	79

Foreword

This European Standard EN 1993-3-1, Eurocode 3: Design of steel structures: Part 3.1: Towers, masts and chimneys – Towers and masts, has been prepared by Technical Committee CEN/TC250 «Structural Eurocodes», the Secretariat of which is held by BSI. CEN/TC250 is responsible for all Structural Eurocodes.

This European Standard shall be given the status of a National Standard, either by publication of an identical text or by endorsement, at the latest by April 2007 and conflicting National Standards shall be withdrawn at latest by March 2010.

This Eurocode supersedes ENV 1993-3-1.

According to the CEN-CENELEC Internal Regulations, the National Standard Organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

Background of the Eurocode programme

In 1975, the Commission of the European Community decided on an action programme in the field of construction, based on article 95 of the Treaty. The objective of the programme was the elimination of technical obstacles to trade and the harmonisation of technical specifications.

Within this action programme, the Commission took the initiative to establish a set of harmonised technical rules for the design of construction works which, in a first stage, would serve as an alternative to the national rules in force in the Member States and, ultimately, would replace them.

For fifteen years, the Commission, with the help of a Steering Committee with Representatives of Member States, conducted the development of the Eurocodes programme, which led to the first generation of European codes in the 1980s.

In 1989, the Commission and the Member States of the EU and EFTA decided, on the basis of an agreement¹ between the Commission and CEN, to transfer the preparation and the publication of the Eurocodes to the CEN through a series of Mandates, in order to provide them with a future status of European Standard (EN). This links *de facto* the Eurocodes with the provisions of all the Council's Directives and/or Commission's Decisions dealing with European standards (e.g. the Council Directive 89/106/EEC on construction products – CPD – and Council Directives 93/37/EEC, 92/50/EEC and 89/440/EEC on public works and services and equivalent EFTA Directives initiated in pursuit of setting up the internal market).

The Structural Eurocode programme comprises the following standards generally consisting of a number of Parts:

- EN 1990 Eurocode 0: Basis of structural design
- EN 1991 Eurocode 1: Actions on structures
- EN 1992 Eurocode 2: Design of concrete structures
- EN 1993 Eurocode 3: Design of steel structures
- EN 1994 Eurocode 4: Design of composite steel and concrete structures
- EN 1995 Eurocode 5: Design of timber structures
- EN 1996 Eurocode 6: Design of masonry structures
- EN 1997 Eurocode 7: Geotechnical design

¹ Agreement between the Commission of the European Communities and the European Committee for Standardisation (CEN) concerning the work on EUROCODES for the design of building and civil engineering works (BC/CEN/03/89).

EN 1998 Eurocode 8: Design of structures for earthquake resistance

EN 1999 Eurocode 9: Design of aluminium structures

Eurocode standards recognise the responsibility of regulatory authorities in each Member State and have safeguarded their right to determine values related to regulatory safety matters at national level where these continue to vary from State to State.

Status and field of application of Eurocodes

The Member States of the EU and EFTA recognise that Eurocodes serve as reference documents for the following purposes :

- as a means to prove compliance of building and civil engineering works with the essential requirements of Council Directive 89/106/EEC, particularly Essential Requirement N°1 - Mechanical resistance and stability - and Essential Requirement N°2 - Safety in case of fire;
- as a basis for specifying contracts for construction works and related engineering services;
- as a framework for drawing up harmonised technical specifications for construction products (ENs and ETAs)

The Eurocodes, as far as they concern the construction works themselves, have a direct relationship with the Interpretative Documents² referred to in Article 12 of the CPD, although they are of a different nature from harmonised product standard³. Therefore, technical aspects arising from the Eurocodes work need to be adequately considered by CEN Technical Committees and/or EOTA Working Groups working on product standards with a view to achieving a full compatibility of these technical specifications with the Eurocodes.

The Eurocode standards provide common structural design rules for everyday use for the design of whole structures and component products of both a traditional and an innovative nature. Unusual forms of construction or design conditions are not specifically covered and additional expert consideration will be required by the designer in such cases.

National Standards implementing Eurocodes

The National Standards implementing Eurocodes will comprise the full text of the Eurocode (including any annexes), as published by CEN, which may be preceded by a National title page and National foreword, and may be followed by a National annex (informative).

The National Annex (informative) may only contain information on those parameters which are left open in the Eurocode for national choice, known as Nationally Determined Parameters, to be used for the design of buildings and civil engineering works to be constructed in the country concerned, i.e. :

- values for partial factors and/or classes where alternatives are given in the Eurocode,
- values to be used where a symbol only is given in the Eurocode,
- geographical and climatic data specific to the Member State, e.g. snow map,
- the procedure to be used where alternative procedures are given in the Eurocode,
- references to non-contradictory complementary information to assist the user to apply the Eurocode.

² According to Art. 3.3 of the CPD, the essential requirements (ERs) should be given concrete form in interpretative documents for the creation of the necessary links between the essential requirements and the mandates for hENs and ETAGs/ETAs.

³ According to Art. 12 of the CPD the interpretative documents should :

- a) give concrete form to the essential requirements by harmonising the terminology and the technical bases and indicating classes or levels for each requirement where necessary ;
- b) indicate methods of correlating these classes or levels of requirement with the technical specifications, e.g. methods of calculation and of proof, technical rules for project design, etc. ;
- c) serve as a reference for the establishment of harmonised standards and guidelines for European technical approvals.

The Eurocodes, *de facto*, play a similar role in the field of the ER 1 and a part of ER 2.

Links between Eurocodes and product harmonized technical specifications (ENs and ETAs)

There is a need for consistency between the harmonised technical specifications for construction products and the technical rules for works⁴. Furthermore, all the information accompanying the CE Marking of the construction products which refer to Eurocodes should clearly mention which Nationally Determined Parameters have been taken into account.

Additional information specific to EN 1993-3-1 and EN 1993-3-2

EN 1993-3 is the third part of six parts of EN 1993 - Design of Steel Structures - and describes the principles and application rules for the safety and serviceability and durability of steel structures for towers and masts and chimneys. Towers and masts are dealt with in Part 3-1; chimneys are treated in Part 3-2.

EN 1993-3 gives design rules in supplement to the generic rules in EN 1993-1.

EN 1993-3 is intended to be used with Eurocodes EN 1990 - Basis of design, EN 1991 - Actions on structures and the parts 1 of EN 1992 to EN 1998 when steel structures or steel components for towers and masts and chimneys are referred to.

Matters that are already covered in those documents are not repeated.

EN 1993-3 is intended for use by

- committees drafting design related product, testing and execution standards;
- clients (e.g. for the formulation of their specific requirements);
- designers and constructors;
- relevant authorities.

Numerical values for partial factors and other reliability parameters in EN 1993-3 are recommended as basic values that provide an acceptable level of reliability. They have been selected assuming that an appropriate level of workmanship and quality management applies.

Annex B of EN 1993-3-1 has been prepared to supplement the provisions of EN 1991-1-4 in respect of wind actions on lattice towers and guyed masts or guyed chimneys.

As far as overhead line towers are concerned all matters related to wind and ice loading, loading combinations, safety matters and special requirements (such as for conductors, insulators, clearance, etc.) are covered by the CENELEC Code EN 50341, that can be referred to for the design of such structures.

The strength requirements for steel members given in this Part may be considered as 'deemed to satisfy', rules to meet the requirements of EN 50341 for overhead line towers, and may be used as alternative criteria to the rules given in that Standard.

Part 3.2 has been prepared in collaboration with Technical Committee CEN/TC 297: Free standing chimneys.

Provisions have been included to allow for the possible use of a different partial factor for resistance in the case of those structures or elements the design of which has been the subject of an agreed type testing programme.

⁴ See Art.3.3 and Art.12 of the CPD, as well as clauses 4.2, 4.3.1, 4.3.2 and 5.2 of ID 1.

National Annex for EN 1993-3-1

This standard gives alternative procedures, values and recommendations for classes with notes indicating where national choices may have to be made. Therefore the National Standard implementing EN 1993-3-1 should have a National Annex containing all Nationally Determined Parameters to be used for the design of buildings and civil engineering works to be constructed in the relevant country.

National choice is allowed in EN 1993-3-1 through paragraphs:

- 2.1.1(3)P
- 2.3.1(1)
- 2.3.2(1)
- 2.3.6(2)
- 2.3.7(1)
- 2.3.7(4)
- 2.5(1)
- 2.6(1)
- 4.1(1)
- 4.2(1)
- 5.1(6)
- 5.2.4(1)
- 6.1(1)
- 6.3.1(1)
- 6.4.1(1)
- 6.4.2(2)
- 6.5.1(1)
- 7.1(1)
- 9.5(1)
- A.1(1)
- A.2(1)P (2 places)
- B.1.1(1)
- B.2.1.1(5)
- B.2.3(1)
- B.2.3(3)
- B.3.2.2.6(4)
- B.3.3(1)
- B.3.3(2)
- B.4.3.2.2(2)
- B.4.3.2.3(1)
- B.4.3.2.8.1(4)
- C.2(1)
- C.6.(1)
- D.1.1(1)
- D.1.2(2)
- D.3(6) (2 places)

EN 1993-3-1:2006 (E)

- D.4.1(1)
- D.4.2(3)
- D.4.3(1)
- D.4.4(1)
- F.4.2.1(1)
- F.4.2.2(2)
- G.1(3)
- H.2(5)
- H.2(7)

1 General

1.1 Scope

1.1.1 Scope of Eurocode 3

See 1.1.1 of EN 1993-1-1.

1.1.2 Scope of Part 3.1 of Eurocode 3

- (1) This Part 3.1 of EN 1993 applies to the structural design of lattice towers and guyed masts and to the structural design of this type of structures supporting prismatic, cylindrical or other bluff elements. Provisions for self-supporting and guyed cylindrical towers and chimneys are given in Part 3.2 of EN 1993. Provisions for the guys of guyed structures, including guyed chimneys, are given in EN 1993-1-11 and supplemented in this Part.
- (2) The provisions in this Part of EN 1993 supplement those given in Part 1.
- (3) Where the applicability of a provision is limited, for practical reasons or due to simplifications, its use is explained and the limits of applicability are stated.
- (4) This Part does not cover the design of polygonal and circular lighting columns, which is covered in EN 40. Lattice polygonal towers are not covered in this Part. Polygonal plated columns (monopoles) may be designed using this Part for their loading. Information on the strength of such columns may be obtained from EN 40.
- (5) This Part does not cover special provisions for seismic design, which are given in EN 1998-3.
- (6) Special measures that might be necessary to limit the consequences of accidents are not covered in this Part. For resistance to fire, reference should be made to EN 1993-1-2.
- (7) For the execution of steel towers and masts, reference should be made to EN 1090.

NOTE: Execution is covered to the extent that is necessary to indicate the quality of the construction materials and products that should be used and the standard of workmanship on site needed to comply with the assumptions of the design rules.

1.2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this European Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this European Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

EN 40	<i>Lighting columns</i>
EN 365	<i>Personal protective equipment against falls from a height. General requirements for instructions for use, maintenance, periodic examination, repair, marking and packaging</i>
EN 795	<i>Protection against falls from a height. Anchor devices. Requirements and testing</i>
EN 1090	<i>Execution of steel structures and aluminium structures</i>
EN ISO 1461	<i>Hot dip galvanized coatings on fabricated iron and steel articles. Specifications and test methods</i>
EN ISO 14713	<i>Protection against corrosion of iron and steel in structures. Zinc and aluminium coatings. Guidelines</i>
ISO 12494	<i>Atmospheric icing of structures</i>
EN ISO 12944	<i>Corrosion protection of steel structures by protective paint systems.</i>

1.3 Assumptions

- (1) See 1.3 of EN 1993-1-1.

1.4 Distinction between principles and application rules

- (1) See 1.4 of EN 1993-1-1.

1.5 Terms and definitions

- (1) The terms and definitions that are defined in EN 1990 clause 1.5 for common use in the Structural Eurocodes apply to this Part 3.1 of EN 1993.

- (2) Supplementary to Part 1 of EN 1993, for the purposes of this Part 3.1, the following definition apply:

1.5.1

global analysis

the determination of a consistent set of internal forces and moments in a structure, that are in equilibrium with a particular set of actions on the structure.

1.5.2

tower

a self-supporting cantilevered steel lattice structure of triangular, square or rectangular plan form, or circular and polygonal monopoles.

1.5.3

guyed mast

a steel lattice structure of triangular, square or rectangular plan form, or a cylindrical steel structure, stabilized at discrete intervals in its height by guys that are anchored to the ground or to a permanent structure.

1.5.4

shaft

the vertical steel structure of a mast.

1.5.5

leg members

steel members forming the main load-bearing components of the structure.

1.5.6

primary bracing members

members other than legs, carrying forces due to the loads imposed on the structure.

1.5.7

secondary bracing members

members used to reduce the buckling lengths of other members.

1.5.8

schiffli-angled angles

modified 90° equal-leg hot rolled angles, each leg of which has been bent to incorporate a 15° bend such that there is an angle of 30° between the outer part of each leg and the axis of symmetry (see Figure 1.1).

1.5.9

wind drag

the resistance to the flow of wind offered by the elements of a tower or guyed mast and any ancillary items that it supports, given by the product of the drag coefficient and a reference projected area, including ice where relevant.

1.5.10**linear ancillary item**

any non-structural components that extend over several panels, such as waveguides, feeders, ladders and pipework.

1.5.11**discrete ancillary item**

any non-structural component that is concentrated within a few panels, such as dish reflectors, aerials, lighting, platforms, handrails, insulators and other items.

1.5.12**projected area**

the shadow area of the element considered, when projected on to an area parallel to the face of the structure normal to the wind direction considered, including ice where relevant. For wind blowing other than normal to one face of the structure, a reference face is used for the projected area. (See Annex A and Annex B.)

1.5.13**panel (of a tower or mast)**

any convenient portion of a tower or mast that is subdivided vertically for the purpose of determining projected areas and wind drag. Panels are typically, but not necessarily, taken between intersections of legs and primary bracings.

1.5.14**section (of a tower or mast)**

any convenient portion of a tower or mast comprising several panels that are nearly or exactly similar, used for the purpose of determining wind drag.

1.5.15**guy**

a tension-only member, connected at each end to terminations to form a guy assembly that provides horizontal support to the mast at discrete levels. The lower end of the guy assembly is anchored to the ground or on a structure and generally incorporates a means of adjusting the tension in the guy.

NOTE 1: Although the terms “stay” and “guy” are generally interchangeable, the word “guy” has been used throughout this document.

NOTE 2: Specific definitions of guys, their make-up and fittings, are provided in Annex D.

1.5.16**damper**

a device that increases the structural damping and thus limits the response of a structure or of a guy.

1.6 Symbols

(1) In addition to those given in EN 1993-1-1, the following main symbols are used:

Latin upper case letters

D_b diameter of the circle through the centre of the bolt hole

D_i diameter of the leg member

G gust response factor

M bending moment

N tension force, number of cycles

N_i number of cycles

N_b axial force

T design life of the structure in years

Latin lower case letters

b	width of a leg of an angle
$c_e(z)$	exposure factor
$c_s c_d$	structural factor
e	eccentricities
h	width of a leg of an angle
k_p	prying effect factor
k_σ	buckling coefficient
m	slope of the S-N curve
n	number of bolts
r_1	radius of the convex part of the bearing
r_2	radius of the concave part of the bearing
t	thickness

Greek upper case letters

ϕ	is the inclination of the mast axis at its base
$\Delta\sigma_E$	stress range

Greek lower case letters

β_A	factor for effective area
χ_M	partial factor
δ_s	logarithmic decrement of structural damping
ε	coefficient depending on f_y
$\bar{\lambda}$	non-dimensional slenderness parameter, equivalence factor
$\bar{\lambda}_p$	non-dimensional slenderness for plate buckling
$\bar{\lambda}_{p,1}$	non-dimensional slenderness parameter for plate buckling of leg 1 of angle
$\bar{\lambda}_{p,2}$	non-dimensional slenderness parameter for plate buckling of leg 2 of angle
ρ	reduction factor

- (2) Further symbols are defined where they first occur.

1.7 Convention for cross section axes

- (1) The convention for axes of angle sections adopted in this Part of EN 1993 is as shown in Figure 1.1.

NOTE: This avoids the confusion inherent in adopting different conventions for hot rolled angles and cold formed angles.

- (2) For built-up members the convention for axes is that of Figure 6.9 of EN 1993-1-1.

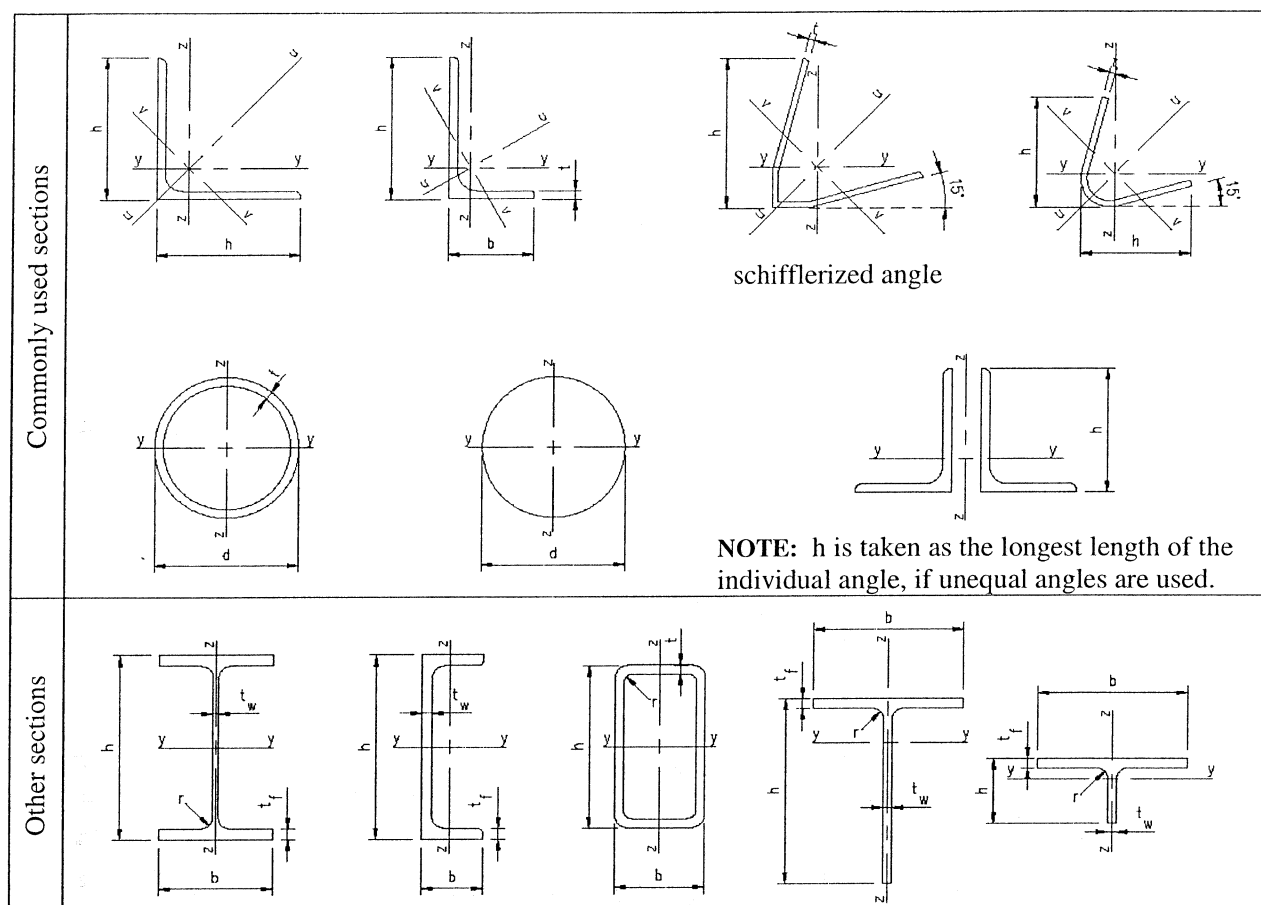


Figure 1.1 Dimensions and axes of sections

2 Basis of design

2.1 Requirements

2.1.1 Basic requirements

- (1)P The design of steel towers and guyed masts shall be in accordance with the general rules given in EN 1990.
- (2) The provisions for steel structures given in EN 1993-1-1 should also be applied.
- (3)P In addition, guyed masts of high reliability (as defined in 2.1.2) shall be designed to withstand the rupture of one guy without collapsing.

NOTE: The National Annex may give information on guy rupture. It is recommended to use the guidance given in Annex E.

2.1.2 Reliability management

- (1) Different levels of reliability may be adopted for the ultimate limit state verifications of towers and masts, depending on the possible economic and social consequences of their collapse.

NOTE: For the definition of different levels of reliability see Annex A.

2.2 Principles of limit state design

- (1) See 2.2 of EN 1993-1-1.

2.3 Actions and environmental influences

2.3.1 Wind actions

- (1) Wind actions should be taken from EN 1991-1-4.

NOTE: The National Annex may give information on how EN 1991-1-4 could be supplemented for masts and towers. The use of the additional rules given in Annex B is recommended.

2.3.2 Ice loads

- (1) Actions from ice should be considered both by their gravity effects and their effect on wind actions.

NOTE: The National Annex may give information on ice loading, the appropriate ice thicknesses, densities and distributions and appropriate combinations, and combination factors for actions on towers and masts. The use of Annex C is recommended.

2.3.3 Thermal actions

- (1) Thermal actions should be determined from EN 1991-1-5 for environmental temperatures.

2.3.4 Selfweight

- (1) Selfweight should be determined in accordance with EN 1991-1-1.
(2) Selfweight of guys should be determined in accordance with EN 1993-1-11.

2.3.5 Initial guy tensions

- (1) The initial guy tensions should be considered as permanent forces, existing in the guys in the absence of meteorological actions, see EN 1993-1-11.
(2) Adjustment for initial guy tensions should be provided. If not, due allowance should be taken in design for the range of initial tensions that might arise, see EN 1993-1-11.

2.3.6 Imposed loads

- (1) Members that are within $[30^\circ]$ to the horizontal should be designed to carry the weight of a workman which for this purpose may be taken as a concentrated vertical load of 1kN.
(2) Imposed loads on platforms and railing should be taken into account.

NOTE 1: The National Annex may give information on imposed loads on platforms and railings. The following characteristic imposed loads are recommended:

- | | | |
|---------------------------------|-----------------------------|------------|
| - Imposed loads on platforms: | 2,0 kN/m² | ... (2.1a) |
| - Horizontal loads on railings: | 0,5 kN/m | ... (2.1b) |

NOTE 2: These loads may be assumed to act in the absence of other climatic loads.

2.3.7 Other actions

- (1) For accidental and collision actions see EN 1991-1-7.

NOTE: The National Annex may give information on the choice of accidental actions.

- (2) Actions during execution should be considered taking due account of the construction scheme. The appropriate load combinations and reduction factors may be obtained from EN 1991-1-6.

NOTE: The limited time for transient design situations may be considered.

- (3) Where considered necessary, actions from settlement of foundations should be assessed. Special considerations may be required for lattice towers founded on individual leg foundations and for differential settlement between the mast base and any guy foundations.

- (4) Actions arising from the fitting and anchoring of safety access equipment may be determined with reference to EN 795. Where the proposed safe method of working requires the use of Work Positioning Systems or mobile fall arrest systems points of attachment should be adequate, see EN 365.

NOTE: The National Annex may give further information.

2.3.8 Distribution of actions

- (1) The loads along the member length including wind or dead loading on other members framing into the member should be considered.

2.4 Ultimate limit state verifications

- (1) For design values of actions and combination factors see EN 1990.

NOTE: For partial factors for actions in the ultimate limit state see Annex A.

- (2) The partial factors for gravity loads and initial tensions in guys should be taken as specified in EN 1993-1-11.

2.5 Design assisted by testing

- (1) The general requirements specified in EN 1990 should be satisfied, in association with the specific requirements given in Section 8 of this Part 3.1 of EN 1993.

NOTE: The National Annex may give further information for structures or elements that are subject to an agreed full-scale testing programme, see 6.1.

2.6 Durability

- (1) Durability should be satisfied by complying with the fatigue assessment (see section 9) and appropriate corrosion protection (see section 4).

NOTE: The National Annex may give information on the design service life of the structure. A service life of 30 years is recommended.

3 Materials

3.1 Structural steel

- (1) For requirements and properties for structural steel, see EN 1993-1-1 and EN 1993-1-3.
- (2) For toughness requirements see EN 1993-1-10.

3.2 Connections

- (1) For requirements and properties for bolts and welding consumables, see EN 1993-1-8.

3.3 Guys and fittings

- (1) For requirements and properties of ropes, strands, wires and fittings see EN 1993-1-11.

NOTE: See also Annex D

4 Durability

4.1 Allowance for corrosion

- (1) Suitable corrosion protection, appropriate to the location of the structure, its design life and maintenance regime, should be provided.

NOTE 1: The National Annex may give further information.

NOTE 2: See also:

- EN ISO 1461 for galvanising,
- EN ISO 14713 for metal spraying and
- EN ISO 12944 for corrosion protection by painting.

4.2 Guys

- (1) For guidance on the corrosion protection of guys see EN 1993-1-11.

NOTE: The National Annex may give further information. The following measures are recommended:

Dependent on the environmental conditions guy ropes made from galvanized steel wires should be given a further layer of protection, such as grease or paint. Care should be taken to ensure that this protective layer is compatible with the lubricant used in the manufacture of the guy ropes.

As an alternate means of protection galvanised steel ropes of diameter up to 20mm may be protected by polypropylene impregnation in which case they do not need further protection unless the sheath is damaged during erection and use. Care needs to be taken in designing the terminations to ensure adequate corrosion protection. Non-impregnated sheathed ropes should not be used because of the risk of corrosion taking place undetected.

Lightning may locally damage the polypropylene coating.

5 Structural analysis

5.1 Modelling for determining action effects

- (1) The internal forces and moments should be determined using elastic global analysis.
- (2) For elastic global analysis see EN 1993-1-1.
- (3) Gross cross-sectional properties may be used in the analysis.
- (4) Account should be taken of the deformation characteristics of the foundations in the design of the structure.
- (5) If deformations have a significant effect (for example towers with large head-loads) second order theory should be used, see EN 1993-1-1.

NOTE 1: Lattice towers may initially be analysed using the initial geometry (first order theory).

NOTE 2: Masts and guyed chimneys should be analysed taking into account the effect of deformations on the equilibrium conditions (second order theory).

NOTE 3: For the overall buckling of symmetric masts see B.4.3.2.6.

- (6) The global analysis of a mast or guyed chimneys should take into account the non-linear behaviour of the guys, see EN 1993-1-11.

NOTE: The National Annex may give further information.

5.2 Modelling of connections

5.2.1 Basis

- (1) The behaviour of the connections should be considered in the global and local analysis of the structure.

NOTE: The procedure for the analysis of connections is given in EN 1993-1-8.

5.2.2 Fully triangulated structures (Simple framing)

- (1) In simple framing the connections between the members may be assumed not to develop moments. In the global analysis, members may be assumed to be effectively pin connected.
- (2) The connections should satisfy the requirements for nominally pinned connections, either:
 - as given in 5.2.2.2 of EN 1993-1-8; or
 - as given in 5.2.3.2 of EN 1993-1-8.

5.2.3 Non-triangulated structures (Continuous framing)

- (1) Elastic analysis should be based on the assumption of full continuity, with rigid connections which satisfy the requirements given in 5.2.2.3 of EN 1993-1-8.

5.2.4 Triangulated structures where continuity is taken into account (continuous or semi-continuous framing)

- (1) Elastic analysis should be based on reliably predicted design moment-rotation or force-displacement characteristics for the connections used.

NOTE: The National Annex may give further information.

6 Ultimate limit states

6.1 General

(1) The following partial factors γ_M apply:

- resistance of member to yielding: γ_{M0}
- resistance of member buckling: γ_{M1}
- resistance of net section at bolt holes: γ_{M2}
- resistance of connections: See Section 6.4
- resistance of guys and their terminations: γ_{Mg} , see EN 1993-1-11
- resistance of insulating material: γ_{Mi}

NOTE 1: The National Annex may give information on partial factors γ_M . The following values are recommended:

$$\gamma_{M0} = 1,00$$

$$\gamma_{M1} = 1,00$$

$$\gamma_{M2} = 1,25$$

$$\gamma_{Mg} = 2,00$$

$$\gamma_{Mi} = 2,50$$

NOTE 2: The factor γ_{Mg} applies to the guy and its socket (or other termination). The associated steel pins, linkages and plates are designed for compatibility with the guy and its socket and may require an enhanced value of γ_{Mg} . For details see EN 1993-1-11.

NOTE 3: For structures or elements that are to be type tested, or where similar configurations have previously been type tested the partial factors, γ_M , may be reduced, subject to the outcome of the testing programme.

6.2 Resistance of cross sections

6.2.1 Classification of cross sections

(1) For towers and masts, classification of cross-sections as given in 5.5.2 of EN 1993-1-1 should be used.

NOTE: The maximum width to thickness ratio c/t for angles defined in table 5.2 of EN 1993-1-1 may be determined with the ratio $(h-2t)/t$ instead of h/t .

6.2.2 Members in lattice towers and masts

(1) For angles connected by one leg, special provisions are given in 3.10.3 of EN 1993-1-8 (if bolted) or 4.13 (if welded).

6.2.3 Guys and fittings

(1) For the strength of guys and fittings see EN 1993-1-11 and Annex D.

6.3 Resistance of members

6.3.1 Compression members

(1) Compression members in lattice towers and masts should be designed using one of the following two procedures:

- a) the method according to the provisions of Annex G and Annex H.
- b) the method given in EN 1993-1-1 taking account of eccentricities.

NOTE 1: The method given in EN 1993-1-1, Annex B. B.1.2(2)B may give conservative results for the buckling resistance of members in lattice towers and masts.

NOTE 2: The choice of the procedure may be made in the National Annex.

- (2) The effective cross section properties of members should be calculated according to 4.3 of EN 1993-1-5.

NOTE 1: For angles the reduction factor ρ may be determined with the slenderness $\bar{\lambda}_p$ taking into account the appropriate width \bar{b} of the compression leg as follows:

- a) for equal leg angles:

$$\bar{\lambda}_p = \frac{\bar{b}/t}{28,4 \varepsilon \sqrt{k_\sigma}} = \frac{(h-2t)/t}{28,4 \varepsilon \sqrt{k_\sigma}}$$

- b) for unequal leg angles:

$$\bar{\lambda}_{p,1} = \frac{\bar{b}/t}{28,4 \varepsilon \sqrt{k_\sigma}} = \frac{(h-2t)/t}{28,4 \varepsilon \sqrt{k_\sigma}} \text{ and}$$

$$\bar{\lambda}_{p,2} = \frac{\bar{b}/t}{28,4 \varepsilon \sqrt{k_\sigma}} = \frac{(b-2t)/t}{28,4 \varepsilon \sqrt{k_\sigma}}$$

NOTE 2: In the case of angles connected by one leg, the reduction factor, ρ , only applies to the connected leg.

NOTE 3: For k_σ see EN 1993-1-5. For a leg of an angle in compression, $k_\sigma = 0,43$.

- (3) The torsional and/or flexural-torsional mode should also be checked as follows:
 - a) Torsional buckling of equal legged angles is covered by the plate buckling verification, see (2).
 - b) For unequal legged angles and all other cross sections, see 6.3.1.4 of EN 1993-1-1 and EN 1993-1-3.
- (4) For cold formed thin gauge members see EN 1993-1-3.

6.4 Connections

6.4.1 General

- (1) For connections see EN 1993-1-8.

NOTE: The partial factors for connections in masts and towers may be given in the National Annex. The numerical values given in Table 2.1 of EN 1993-1-8 are recommended.

- (2) All bolts should be secured against loosening.

6.4.2 Tension bolts in end plates (flanged connections)

- (1) Where there is a possibility of tension across the flange connection preloaded bolts should be used.
- (2) The minimum bolt diameter should be 12mm.

NOTE: The National Annex may give further information on flange connections of circular hollow sections and cylindrical shells. For circular hollow sections the following simplified method for members in tension without bending is recommended, see figure 6.1.

In determining the flange thickness the following is relevant:

- the shear resistance of the flange along the perimeter of the connected circular leg section;
- the resistance to combined bending and shear of the flange along the circle through the bolt holes. The bending moment (M) may be taken as:

$$M = N (D_b - D_i)/2$$

where: N is the tension force in the leg member

D_b is the diameter of the circle through the centre of the bolt holes

D_i is the diameter of the leg member

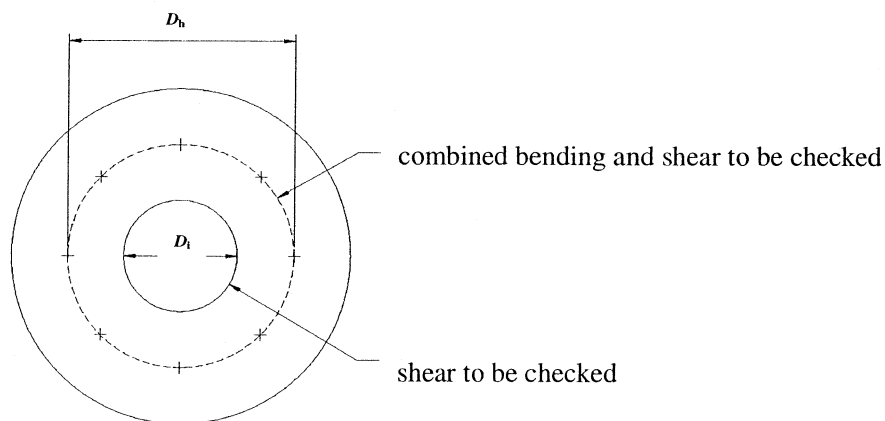


Figure 6.1 Bolted flanged connections

In determining the forces in the bolts, the axial force N_b

$$N_b = \frac{N k_p}{n}$$

where: n is the number of bolts

k_p is a prying effect factor taken as

$k_p = 1,2$ for pre-loaded bolts

$= 1,8$ for non-preloaded bolts

All bolts should be preloaded for fatigue, see EN 1993-1-8

6.4.3 Anchor bolts

- (1) Where fatigue needs to be considered anchor bolts should be preloaded. In such cases appropriate steel materials should be used, see EN 1993-1-8.

NOTE: For the choice of the preload see also rules for prying force eccentricity, stress levels etc. in EN 1993-1-8.

6.4.4 Welded connections

- (1) See EN 1993-1-8.

NOTE: For execution see EN 1090.

6.5 Special connections for masts

6.5.1 Mast base joint

- (1) The design bearing stress on the spherical pinned connection should be based on the design rules for rocker bearings, see EN 1337-6.

NOTE: The National Annex may give information on eccentricities and limit values for the Hertz pressure.

To verify that the area of the compression zone is within the boundaries of the bearing parts taking due account of the true rotation angle of the mast base section (see Figure 6.2) and to determine the bending moments caused by the resulting eccentricities for designing the bearing and the bottom section of the mast the following rules for determining eccentricities are recommended:

If the mast base rests on a spherical bearing the point of contact should be assumed to move in the direction of any inclination of the mast axis by rolling over the bearing surface.

The eccentricities e_u and e_o (see Figure 6.2) should be determined as follows:

$$e_u = r_1 \times \sin \psi_1 \quad \dots (6.12a)$$

$$e_o = r_2 (\sin \psi_1 - \sin \phi) \quad \dots (6.12b)$$

where: r_1 is the radius of the convex part of the bearing;

r_2 is the radius of the concave part of the bearing;

and $r_2 > r_1$

ϕ is the inclination of the mast axis at its base.

$$\text{with: } \psi_1 = \frac{r_2 \phi}{r_2 - r_1} \quad \dots (6.13a)$$

$$\psi_2 = \psi_1 - \phi \quad \dots (6.13b)$$

If r_2 is infinite, that is a flat surface, then e_o should be taken as $e_o = r_1 \phi \cos \phi$.

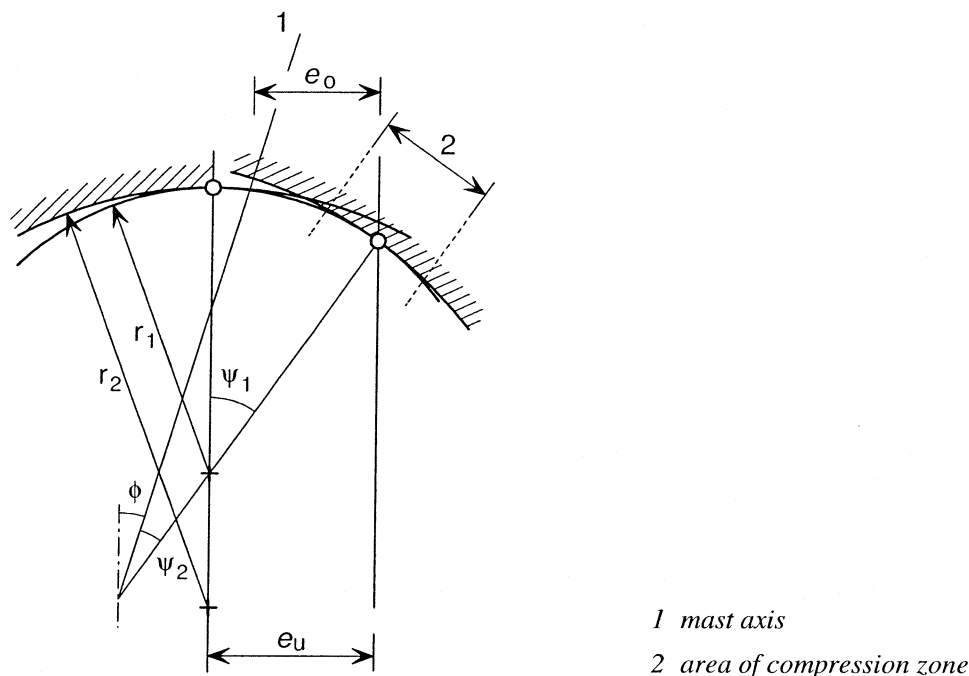


Figure 6.2 Eccentricities due to the inclination of the mast base

(2) The system for suppressing twisting of a pinned mast base joint should be designed to permit rotation of the mast base section around the horizontal axes.

(3) For a fixed mast base possible settlements of the shaft foundation and of the guy foundations should be considered in the mast design.

6.5.2 Guy connections

(1) All connections of the guys to the mast or to guy foundations should allow the guy to rotate freely in both vertical and horizontal directions, see EN 1993-1-11.

Account should be taken in the design and detailing connections of the tendency for guy constructions to twist under tensile loading.

NOTE: Generally for connections with pins the freedom for horizontal rotations can be obtained by a “spherical” form of the hole in the centre plate for the pin. Spherical bearings may be used in exceptional circumstances.

(2) All pins should be adequately secured against lateral movement by the use, for example, of a nut combined with a split pin.

(3) The guy attachment plate in the mast and the steel anchor plate projecting from the guy foundation should both be designed for the lateral force from the guy due to the wind loading component normal to the plane of the guy.

(4) Wherever practicable welded connections should be detailed to enable visual and non-destructive inspections to be undertaken in service.

7 Serviceability limit states

7.1 Basis

- (1) The following serviceability limit states may be relevant for design:
- deflections or rotations that adversely affect the effective use of the structure, including the proper functioning of aerials or services;
 - vibration, oscillation or sway that causes loss of transmitted signals;
 - deformations, deflections, vibration, oscillation or sway that causes damage to non-structural elements.

NOTE: The National Annex may give information on limits and associated χ_M -values. The value $\chi_M = 1,0$ is recommended.

7.2 Deflections and rotations

7.2.1 Requirements

- (1) The maximum deflections and rotations should be determined using the combination of characteristic actions on the structure and its ancillaries.
- (2) The deflections and rotations for masts and guyed chimneys should be calculated making due allowance for any second order effects, see EN 1993-1-1, and any dynamic effects.

7.2.2 Definition of limiting values

- (1) Limiting values should be specified together with the load case considered.

NOTE: For guyed masts see Annex B.

- (2) For broadcasting and floodlighting structures, the limiting values to be considered should be taken as those for horizontal displacement and rotation at the top of the structure. For directional antennae the limiting values should be taken at the point of the attachment of the directional antenna.

7.3 Vibrations

- (1) Towers and masts should be examined for:
- gust induced vibrations (causing vibrations in the direction of the wind);
 - vortex induced vibrations for towers or masts containing prismatic cylindrical or bluff elements or shrouds (causing vibrations perpendicular to the direction of the wind);
 - galloping instability (causing vibrations of the guys);
 - rain-wind induced vibrations.

NOTE 1: For dynamic effects see EN 1991-1-4 and Annex B and also Annex B of EN 1993-3-2.

NOTE 2: Vibrations can cause rapid development of fatigue damage, see section 9.

- (2) If lattice towers and masts or guyed chimneys are predicted to be subject to wind vibrations, unless other measures are taken to reduce these in the design, provisions should be made for the installation of damping devices if found necessary in the light of experience.

NOTE: See Annex B of EN 1993-3-2.

8 Design assisted by testing

- (1) The provisions for design assisted by testing given in EN 1990 should be followed.
- (2) Where the values of the logarithmic decrement of structural damping, δ_s given in EN 1991-1-4 are considered inappropriate for lattice towers and masts consisting of, supporting or containing cylindrical elements, testing may be undertaken to determine these values.

NOTE: Guidance for the determination of δ_s is given in Annex C to EN 1993-3-2.

- (3) Higher modes than the fundamental might be significant, particularly for guyed masts, so due account of this should be taken in determining the appropriate logarithmic decrement of structural damping.
- (4) Account should be taken of the fact that the frequencies of vibration vary according to the loading condition considered for instance in still air, under wind, or under ice loading.

9 Fatigue

9.1 General

- (1) For fatigue verifications the provisions of EN 1993-1-9 should be applied.
- (2) Consideration should be given to the effects on fatigue resistance of the possible existence of secondary moments in lattice towers and masts that are not already allowed for.

9.2 Fatigue loading

9.2.1 In-line vibrations

- (1) Fatigue loading of lattice towers and guyed masts due to in-line vibrations (without cross-wind vibrations) induced by gusty wind need not be determined provided that the detail category of the structural details are greater than 71 N/mm².

NOTE: The fatigue life of structures subject to in-line vibrations only (without cross-wind vibrations) induced by gusty wind may be assumed to be greater than 50 years.

- (2) In all other cases due account should be taken of the details adopted, and fatigue verification undertaken.

NOTE: For the fatigue verification due to in-line vibrations see EN 1991-1-4. The following simplified method may be used:

- a) The fatigue stress history due to wind gusts is evaluated by determining the annual durations of different mean wind speeds from different directions from meteorological records for the site. The fluctuations about the mean values may then be assumed to have a statistically normal distribution with a standard deviation in stress corresponding to $G/4$ times the stress due to the mean wind speed. The appropriate gust response factor G is defined as

$$G = c_e(z) c_s c_d - 1$$

where

$c_e(z)$ is the exposure factor, see EN 1991-4

$c_s c_d$ is the structural factor, see EN 1991-4

derived in accordance with Annex B.

- b) The stress range, $\Delta\sigma_{Si}$, may be assumed to be 1,1 times the difference between the stress arising from that incorporating the gust response factor G and that due to the ten minute mean wind speed. An equivalent number of cycles N_i may then be obtained from:

$$N_i = 10^5 T/50 \quad \dots (9.1)$$

where: T is the design life of the structure in years.

9.2.2 Cross-wind vortex vibrations

- (1) The fatigue loading of towers and guyed masts consisting of, supporting or containing prismatic, cylindrical or other bluff elements should be determined from the maximum amplitude for the relevant vibration mode and the number of stress cycles N .

NOTE: For the fatigue actions see EN 1991-1-4, Annex E.

9.2.3 Individual member response

- (1) Slender individual members of structures should be assessed for cross-wind excitation.

NOTE: For the fatigue actions see EN 1991-1-4, Annex E. The limitations on slenderness given in Annex H H.2(1) and H.3.1(3) will generally be sufficient to prevent such excitation. An increase of damping (friction, additional dampers) is a practical means of suppressing such vibrations if they occur in practice.

9.3 Fatigue resistance

- (1) Reference should be made to EN 1993-1-9 which includes resistances of details typical for towers, chimneys and guyed masts.

9.4 Safety assessment

- (1) The safety assessment for fatigue should be carried out in accordance with 8(2) of EN 1993-1-9, using:

$$\Delta\sigma_{E2} = \lambda \Delta\sigma_E \quad \dots (9.2)$$

where:

λ is the equivalence factor to transfer $\Delta\sigma_E$ to $N_c = 2 \times 10^6$ cycles;

$\Delta\sigma_E$ is the stress range associated to N cycles (see 9.2) allowing for stress concentration factors where appropriate.

- (2) The equivalence factor λ may be determined from:

$$\lambda = \left(\frac{N}{2 \times 10^6} \right)^{\frac{1}{m}} \quad \dots (9.3)$$

where: m is the slope of the S-N curve.

9.5 Partial factors for fatigue

- (1) The partial factors for fatigue should be taken as specified in 3(6) and (7) and 6.2(1) of EN 1993-1-9.

NOTE: The National Annex may give numerical values for γ_{Ff} and γ_{Mf} . For γ_{Ff} the value $\gamma_{Ff} = 1,00$ is recommended. For γ_{Mf} values see Table 3.1 in EN 1993-1-9.

9.6 Fatigue of guys

- (1) The fatigue performance of guys should be verified using the procedures given in EN 1993-1-11.

Annex A [normative] – Reliability differentiation and partial factors for actions

NOTE: As this Annex deals with reliability differentiation and partial factors for actions for masts and towers, it is expected that it will be transferred to Annex A to EN 1990 in a later stage.

A.1 Reliability differentiation for masts and towers

- (1) Reliability differentiation may be applied to masts and towers by the application of reliability classes.

NOTE: The National Annex may give relevant reliability classes related to the consequences of structural failure. The classes in Table A.1 are recommended.

Table A.1 Reliability differentiation for towers and masts

Reliability Class	
3	towers and masts erected in urban locations, or where their failure is likely to cause injury or loss of life; towers and masts used for vital telecommunication facilities; other major structures where the consequences of failure would be likely to be very high
2	all towers and masts that cannot be defined as class 1 or 3
1	towers and masts built on unmanned sites in open countryside; towers and masts, the failure of which would not be likely to cause injury to people

A.2 Partial factors for actions

- (1)P Partial factors for actions shall be dependant on the reliability class of the tower or mast.

NOTE 1: In the choice of partial factors for permanent actions γ_G and for variable actions γ_Q the dominance of wind actions for the design may be taken into account.

NOTE 2: The National Annex may give numerical values of γ_G and γ_Q . Where the reliability classes recommended in A.1 are used the numerical values in Table A.2 for γ_G and γ_Q are recommended.

Table A.2 Partial factors for permanent and variable actions

Type of Effect	Reliability Class, see NOTE to 2.1.2	Permanent Actions	Variable Actions (Q_s)
unfavourable	3	1,2	1,6
	2	1,1	1,4
	1	1,0	1,2
favourable	All Classes	1,0	0,0
Accidental situations		1,0	1,0

NOTE 3: The National Annex may also give information on the use of dynamic response analysis for wind actions, see Annex B.

Annex B [informative] – Modelling of meteorological actions

NOTE: As this Annex deals with supplementary rules for wind actions on lattice towers, guyed masts and guyed chimneys, and on their response, it is expected that it will be transferred to EN 1991-1-4 in a later stage.

B.1 General

B.1.1 Scope of this Annex

(1) This Annex contains supplementary information about wind actions on towers and guyed masts as follows:

- wind force, see B.2;
- response of lattice towers, see B.3; and
- response of guyed masts, see B.4.

NOTE: This Annex refers to ISO 12494 for ice loading. The National Annex may give further information.

B.1.2 Symbols

(1) In addition to those given in EN 1993-1-1 and EN 1991-1-4, the following main symbols have been used in this Annex:

i	patch load pattern
K	factor
L	projected length, chord length
N	number
Q	parameter
S	load effect in a member (e.g. force, shear or bending moment)
T	torque
α	slope of guy to horizontal
β	parameter
η	shielding factor
θ	angle of wind incidence to the normal in plane; slope
τ	constant
ψ	angle of wind incidence to the longitudinal axis
ω	spacing ratio
k_s	scaling factor

(2) In addition to those given in EN 1993-1-1 the following subscripts have been used in this Annex:

A	ancillary item
C	cantilever
c	circular-section members
e	effective
F	face
f	flat-sided members
G	guy
H	mast height
L	length
M	base mast or mast only
m	mast; mean
n	single frame
PL	patch load
p	patch
q	shear
S	structure

sup	super-critical
T	tower, total
w	in the direction of the wind
w	with wind
x	in the crosswind direction
Z	in the vertical direction
z	height z above ground level
θ	angle of wind incidence

B.2 Wind force

B.2.1 General

B.2.1.1 Outline

- (1) For the purposes of calculating the wind force, the structure should be divided into a series of sections, where a section comprises several identical or nearly identical panels, see Figure B.2.1. Projections of bracing members in faces parallel to the wind direction, and in plan and hip bracing, should be omitted in the determination of the projected area of the structure.
- (2) The structure should generally be divided into a sufficient number of sections to enable the wind loading to be adequately modelled for the global analysis.
- (3) The wind force acting on a section or component should be determined according to 5.3 (2) of EN 1991-1-4.
- (4) In determining the wind force under iced conditions, the projected areas of structural elements and ancillaries should be increased to take due account of the thickness of ice as relevant.
- (5) In applying the method given in this Annex, the maximum force within an angle of $\pm 30^\circ$ to the nominal wind direction should be used to obtain the maximum loading in the wind direction.

NOTE: The National Annex may give information of wind tunnel tests.

B.2.1.2 Method

- (1) The method given in B.2.1.3 should be used to determine the wind force on square or equilateral triangular lattice structures.

NOTE 1: The procedure given in B.2.7 only applies for either:

- a) as guidance for structures of rectangular cross section; or
- b) the assessment of existing structures for which the disposition of ancillaries and aerals is accurately known.

NOTE 2: The procedure given in B.2.7 may provide lower values of drag than the method given in B.2.1.3 when K_A is taken as 1,0 in B.2.3 and B.2.4.

B.2.1.3 Total wind force coefficient

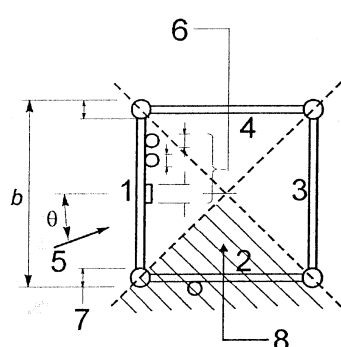
(1) The total wind force coefficient c_f in the direction of the wind over a section of the structure should be taken as:

$$c_f = c_{f,S} + c_{f,A} \quad \dots (B.1)$$

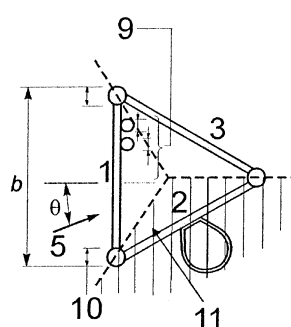
where: $c_{f,S}$ is the wind force coefficient of the bare structure section, determined in accordance with B.2.2 using the solidity ratio, ϕ , appropriate to the bare structure; and

$c_{f,A}$ is the wind force coefficient of the ancillaries, determined in accordance with B.2.3 and B.2.4, as appropriate.

(2) Where the projected areas of ancillaries on each face are within 10% of each other, then they may be treated as appropriate structural members and the total wind drag calculated in accordance with B.2.2.



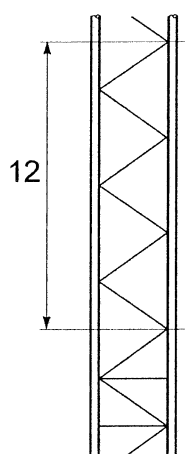
NOTE: Face 1 should be taken as windward such that $-45^\circ \leq \theta \leq 45^\circ$



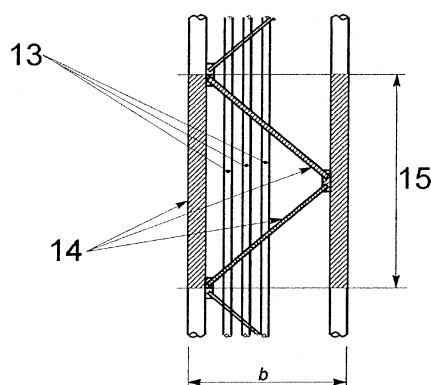
NOTE: Face 1 should be taken as windward such that $-60^\circ \leq \theta \leq 60^\circ$. External ladder should be treated as individual item.

(a) Plan on square structure

(b) Plan on triangular structure



(c) Mast section



(d) Structural panel

- 1 Face 1
- 2 Face 2
- 3 Face 3
- 4 Face 4
- 5 Wind
- 6 Ancillary components projected normal to face 1
- 7 Leg projected normal to face
- 8 Ancillary components in this area allocated to face 2
- 9 Ancillary components projected normal to face (inclusive of ladder rungs, hoops etc.)
- 10 Leg projected normal to face
- 11 Ancillary components in this area allocated to face 2
- 12 Mast section
- 13 Ancillary components of projected area A_A
- 14 Structural components of projected area A_S
- 15 Panel height (h)

Figure B.2.1 Projected panel area used to calculate solidity ratio, ϕ

B.2.2 Wind force coefficient of structural components

B.2.2.1 General

(1) For a lattice structure of square or equilateral triangular plan form, having equal areas on each face, the total wind force coefficient $c_{f,S}$ of a section in the direction of the wind:

$$c_{f,S} = K_{\theta} c_{f,S,0} \quad \dots (B.2)$$

where: $c_{f,S,0}$ is the overall normal drag (pressure) coefficient of a section j without end-effects, determined in accordance with B.2.2.2;

K_{θ} is the wind incidence factor.

(2) The wind incidence factor K_{θ} may be obtained from:

$$K_{\theta} = 1,0 + K_1 K_2 \sin^2 2\theta \text{ for square structures} \quad \dots (B.3a)$$

$$K_{\theta} = \frac{A_c + A_{c,sup}}{A_S} + \frac{A_f}{A_S} (1 - 0,1 \sin^2 1,5\theta) \text{ for triangular structures} \quad \dots (B.3b)$$

$$\text{with: } K_1 = \frac{0,55 A_f}{A_S} + \frac{0,8(A_c + A_{c,sup})}{A_S} \quad \dots (B.3c)$$

$$K_2 = 0,2 \text{ for } 0 \leq \varphi \leq 0,2 \text{ and } 0,8 \leq \varphi \leq 1,0 \quad \dots (B.3d)$$

$$= \varphi \text{ for } 0,2 < \varphi \leq 0,5 \quad \dots (B.3e)$$

$$= 1 - \varphi \text{ for } 0,5 < \varphi < 0,8 \quad \dots (B.3f)$$

in which: θ is the angle of incidence of the wind to the normal of face 1, in plan

φ is the solidity ratio see 7.11 (2) of EN 1991-1-4.

A_f is the total projected area when viewed normal to the face of the flat-sided section members in the face

A_c is the total projected area when viewed normal to the face of the circular-section members in the face in sub critical regimes

$A_{c,sup}$ is the total projected area when viewed normal to the face, of the circular-section members in the face in supercritical regimes

h is the section height under consideration

b is the overall section width, as shown in Figure B.2.1

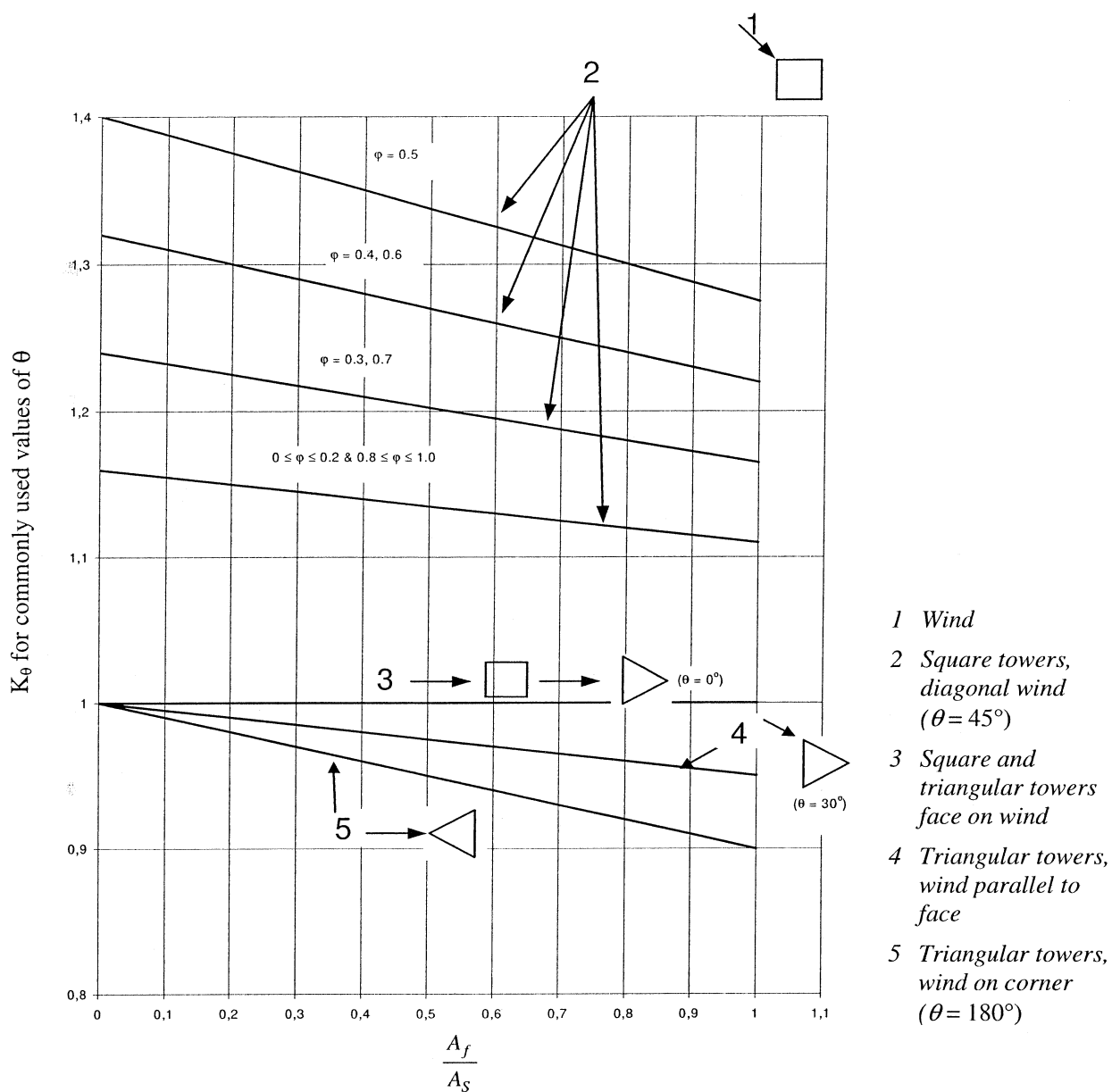
NOTE: $A_S = A_f + A_c + A_{c,sup}$

(3) Values of K_{θ} for commonly used values of θ may be obtained from Figure B.2.2.

(4) Circular-section members should be assumed to be in a sub critical regime when the effective Reynold's number $Re \leq 4 \times 10^5$, and may be assumed to be in a supercritical regime for higher values of Re only when they are ice free.

(5) The value of Re should be obtained from 7.9.1(1) of EN 1991-1-4.

(6) Where supercritical flow is assumed for any or all members, it should be checked that greater loading does not result under a reduced wind speed corresponding to $Re < 4 \times 10^5$.

Figure B.2.2 Wind incidence factor K_θ

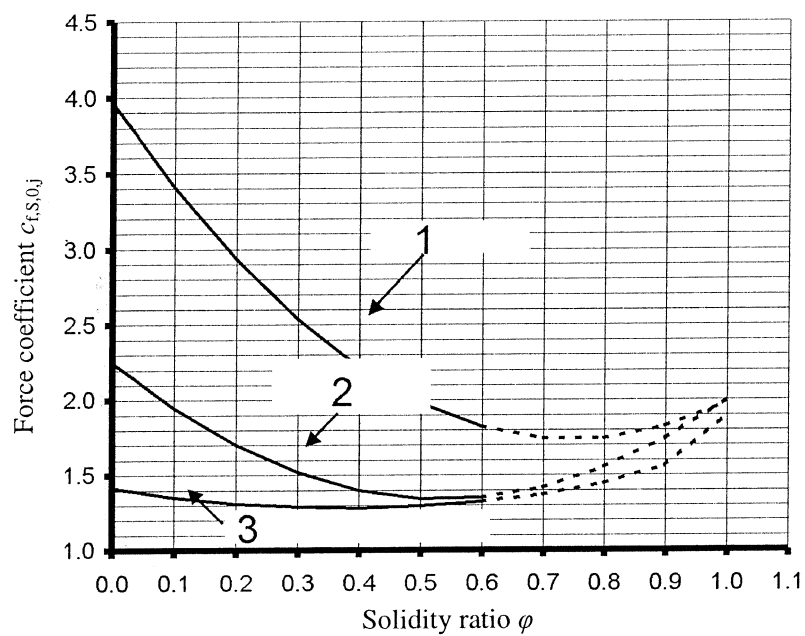
$$c_{f,s,0,j} = c_{f,0,f} \frac{A_f}{A_s} + c_{f,0,c} \frac{A_c}{A_s} + c_{f,0,c,\text{sup}} \frac{A_{c,\text{sup}}}{A_s} \quad \dots \text{ (B.4)}$$
$$c_{f,0,f} = 1,76 C_1 [1 - C_2 \varphi + \varphi^2] \quad \dots (B.5a)$$

$$c_{f,0,c} = C_1 (1 - C_2 \varphi) + (C_1 + 0,875) \varphi^2 \quad \dots (B.5b)$$

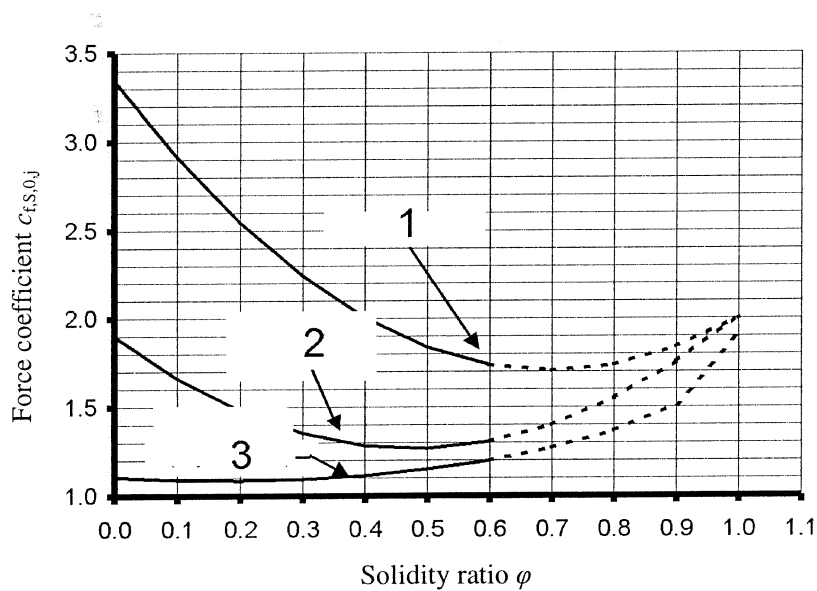
$$c_{f,0,c,\sup} = 1,9 - \sqrt{\{(1-\phi)(2,8-1,14C_1+\phi)\}} \quad \dots (B.5c)$$

where: $\varphi, A_S, A_f, A_c, A_{c,\text{sup}}$ are defined in B.2.2.1

(3) Approximate values of these force coefficients may be obtained from Figure B.2.3.



(a) Square structures



(b) Triangular structures

- 1 Flat-sided
- 2 Circular (sub critical)
- 3 Circular (supercritical)

NOTE: For structures with $\phi > 0,6$ consideration should be given to the possibility of cross-wind response due to vortex excitation, see EN 1991-1-4.

Figure B.2.3 Overall normal force coefficients $c_{t,s,0}$ for square and triangular structures

B.2.3 Wind force coefficients of linear ancillaries

(1) The wind force coefficient $c_{f,A}$ in the direction of the wind of any linear ancillary part (including waveguides, feeders, etc.) within a panel height should be taken as:

$$c_{f,A} = c_{f,A,0} K_A \sin^2 \psi \quad \dots (B.6)$$

where: $c_{f,A,0}$ is the overall normal drag coefficient appropriate to the item and its effective Reynold's number, values of which are given in table B.2.1 for common isolated individual members and may be determined in accordance with B.2.7.2 for parts composed of single frames;

K_A is a reduction factor to take account of the shielding of the component by the structure itself and may only be taken into account when at least one face of the structure is effectively shielding the component (or vice versa); K_A is given in table B.2.2 except for circular sections in supercritical flow and for ancillaries not complying with the constraints of B.2.3 (2) in which case $K_A = 1.0$;

NOTE: Where A_A is greater than A_S the reduction factor should be applied to $c_{f,S,0}$ rather than $c_{f,A}$. Thus in such cases:

$$c_{f,S} = K_\theta c_{f,S,0} K_A$$

$$c_{f,A} = c_{f,A,0} \sin^2 \psi$$

ψ is the angle of wind incidence to the longitudinal axis of any linear member.

(2) K_A should be taken as 1,0 for ancillary items that do not conform to any of the following restraints:

- the total projected area of those ancillary parts adjacent to the face under consideration is less than the projected area of the structural members in that face (see Figure B.2.1);
- the total projected area normal to any face on the structure of any single internal or external ancillary is less than half the gross area of the face of the panel (see figure B.2.1);
- any ancillary does not extend more than 10% beyond the total face width of the structure at that level.

Table B.2.1 Typical force coefficients, $c_{f,A,0}$ and $c_{f,G}$, for individual components

Member type	Effective Reynold's number Re (see EN 1991-1-4) (see NOTE 1)	Drag (pressure) coefficient $c_{f,A,0}$ or $c_{f,G}$	
		Ice-free	Iced
(a) Flat-sided sections and plates	All values	2,0	2,0
(b) Circular sections and smooth wire	$\leq 2 \times 10^5$	1,2	1,2
	4×10^5	0,6	1,0
	$> 10 \times 10^5$	0,7	1,0
(c) Fine stranded cable, e.g. steel core aluminium round conductor, locked coil ropes, spiral steel strand with more than seven wires	Ice free:		
	$\leq 6 \times 10^4$	1,2	
	$\geq 10^5$	0,9	
	Iced:		
	$\leq 1 \times 10^5$		1,25
	$\geq 2 \times 10^5$		1,0
(d) Thick stranded cable, e.g. small wire ropes, round strand ropes, spiral steel strand with seven wires only (1 x 7)	Ice free:		
	$\leq 4 \times 10^4$	1,3	
	$> 4 \times 10^4$	1,1	
	Iced:		
	$\leq 1 \times 10^5$		1,25
	$\geq 2 \times 10^5$		1,0
(e) Cylinders with helical strakes of depth up to 0.12D (see NOTE 2)	All values	1,2	1,2

NOTE 1: For intermediate values of Re, $c_{f,A,0}$ should be obtained by linear interpolation.
NOTE 2: These values are based on the overall width, including twice the strake depth.
NOTE 3: The values for iced components are relevant for glazed ice; care should be exercised if they are used for rime ice (see ISO 12494).
NOTE 4: These values may be changed in the National Annex.

- (3) Where relevant, the corresponding torsional force T_{AW} should be calculated using the appropriate coefficient obtained from wind tunnel tests with the relevant moment arm for such torsion.

Table B.2.2 Reduction factor, K_A , for ancillary items

Position of ancillaries	Reduction factor, K_A	
	Square or rectangular plan form	Triangular plan form
Internal to the section	0,8	0,8
External to the section	0,8	0,8

NOTE: These values may be changed in the National Annex.

B.2.4 Wind force coefficients of discrete ancillaries

- (1) For any discrete ancillary item such as a dish reflector, the total wind force coefficient $c_{f,A}$ in the direction of the wind, should be taken as:

$$c_{f,A} = c_{f,A,0} K_A \quad \dots (B.7)$$

where: $c_{f,A,0}$ is the force coefficient for the item appropriate to the wind direction and wind speed and should be obtained from wind tunnel tests generally provided by the manufacturer;

K_A is as defined in B.2.3.

- (2) The corresponding crosswind force coefficients $c_{f,A,x}$ and lift coefficient $c_{l,A,z}$ should be calculated as for $c_{f,A}$ taking the reference direction in plan as normal to the mean wind direction, and $c_{f,A,0}$ as the appropriate coefficient for crosswind and lift.

- (3) The corresponding torsional force coefficient T_{AW} should be calculated using the appropriate coefficient, obtained from wind tunnel tests in association with the relevant moment arm for such torsion.

B.2.5 Wind force coefficients of guys

- (1) The wind force coefficient $c_{f,G}$ normal to the guys in the plane containing the guy and the wind should be taken as:

$$c_{f,G} = c_{f,G,0} \sin^2 \psi \quad \dots (B.8)$$

where: $c_{f,G}$ is the overall normal drag coefficient appropriate to the effective Reynold's number, the values of which are given in table B.2.1 for both ice-free and iced conditions;

ψ is the angle of wind incidence to the chord.

NOTE: The wind force on guy insulators, where relevant, should be accounted for, either by using their appropriate wind force coefficients as individual elements along the guy, or by smearing their effect into $c_{f,G}$.

B.2.6 Wind force coefficients under iced conditions

- (1) In determining the wind resistance of a structure and ancillaries under iced conditions, each element of the structure, ancillary parts and guys should be taken as coated on all sides by ice, with a thickness equal to that given in Annex C.

- (2) Where the gap between components when not iced, is less than 75 mm, this should be assumed to be completely filled by ice under icing conditions.

- (3) Force coefficients of individual members should be obtained from table B.2.1.

- (4) Consideration should be given to asymmetric ice in which some guys are iced and some are ice-free (see Annex C).

B.2.7 Guidance for special cases

B.2.7.1 Total wind force coefficient

- (1) The total wind force coefficient c_f in the direction of the wind over a panel height of a square or triangular structure or of a structure of rectangular unequal sided cross-section may be determined from (2) below.

NOTE: For the design of square or equilateral triangular structures the method given in B.2.1.3 should be used.

- (2) The total wind force coefficient, c_f , in the direction of the wind over a panel height may be determined as follows:

- for square and rectangular structures:

$$c_f = c_{le} \cos^2 \theta_1 + c_{2e} \sin^2 \theta_1 \quad \dots (B.9)$$

- for triangular structures:

$$c_f = c_{le} \cos^2 \left(\frac{3\theta_1}{4} \right) + c_{2e} \sin^2 \left(\frac{3\theta_1}{4} \right) \quad \dots (B.10)$$

where: c_{le} is an effective wind force coefficient given by the following:

- for square and rectangular structures:

$$c_{le} = (c_1 + \eta_1 c_3) K_{\theta 1}$$

- for triangular structures:

$$c_{le} = \left\{ c_1 + \frac{\eta_1}{2} (c_2 + c_3) \right\} K_{\theta 1}$$

c_{2e} is an effective wind force coefficient given by the following:

- for square and rectangular structures:

$$c_{2e} = (c_2 + \eta_2 c_4) K_{\theta 2}$$

- for triangular structures:

$$c_{2e} = \left\{ c_2 + \frac{\eta_2}{2} (c_1 + c_3) \right\} K_{\theta 2}$$

c_1 to c_4 are wind force coefficients given by:

$$c_1 = c_{f,S1} A_{S1}/\Sigma A + c_{f,A1} A_{A1}/\Sigma A;$$

$$c_2 = c_{f,S2} A_{S2}/\Sigma A + c_{f,A2} A_{A2}/\Sigma A;$$

$$c_3 = c_{f,S3} A_{S3}/\Sigma A + c_{f,A3} A_{A3}/\Sigma A;$$

$$c_4 = c_{f,S4} A_{S4}/\Sigma A + c_{f,A4} A_{A4}/\Sigma A;$$

A_{S1} to A_{S4} are the areas projected normal to faces 1, 2, 3 and 4, respectively, of the components treated as structural members within the same panel height of faces 1, 2, 3 and 4 including icing, where appropriate (see Figure B.2.1);

A_{A1} to A_{A4} are the areas projected normal to the faces 1, 2, 3 and 4 respectively of the ancillary items within the same panel height of faces 1, 2, 3, 4 including icing where appropriate (see Figure B.2.1).

$c_{f,S1}$ to $c_{f,S4}$ are the force coefficients appropriate to faces 1 to 4, respectively, of the components treated as structural members which may be determined in accordance with B.2.7.2;

$c_{f,A1}$ to $c_{f,A4}$ are the wind force coefficients appropriate to faces 1 to 4, respectively, for the ancillary items not treated as structural members, determined in accordance with B.2.3 or B.2.4, as appropriate but taking $K_A = 1,0$ in all cases;

η_1 and η_2 are the effective shielding factors for faces 1 and 2, respectively, including both structural and ancillary components.

- for square structures η_1 and η_2 should be taken as: η_e
- for triangular structures η_1 and η_2 should be taken as: $0,67 \eta_e$
- for rectangular structures η_1 and η_2 should be taken as: $\eta_e + 0,15(\omega - 1)(\varphi - 0,1)$ but not greater than 1,0

$$\eta_e = \eta_f (A_f + 0,83 A_c + 2,1 A_{c,sup} + A_A) / (A_s + A_A) \text{ but not greater than } 1,0;$$

$$\eta_f \text{ is given by: } \eta_F = (1 - \varphi)^{1,89} \text{ and is plotted in Figure B.2.4}$$

where: A_f , A_c , $A_{c,sup}$ are as defined in B.2.2.1 applicable to faces 1 or 2, as appropriate;

$$A_s = A_f + A_c + A_{c,sup} \text{ (see B.2.2.1(2))}$$

A_A is the projected area normal to the face of the ancillary items not treated as structural members applicable to faces 1 to 4, as appropriate;

φ is the solidity ratio appropriate to face 1 or 2, as defined in Figure B.2.2, but including both structural and ancillary components

$$\text{Thus } \varphi = \frac{A_s + A_A}{hb}$$

ω is the spacing ratio for rectangular structures, equal to the distance between the face considered and that parallel to it divided by the width of the face considered at the level of the centroid of the panel area but not to be taken as less than 1,0;

$K_{\theta 1}$ and $K_{\theta 2}$ are to be obtained from B.2.2.1, applicable to faces 1 or 2, as appropriate, using $(A_s + A_A) A_f$ and φ as defined in this subclause;

θ_1 is the plan angle of incidence of wind to the normal to face 1.

EN 1993-3-1:2006 (E)

- (3) For structures with $\varphi > 0.6$ consideration should be given to the possibility of cross-wind response due to vortex excitation, see EN 1991-1-4.

- (4) The total crosswind force coefficients over a panel c_{fx} should be determined as in (2), but taking the reference direction as normal in plan to the mean wind direction.
- (5) The total wind force coefficient, c_f , in the direction of the wind over a panel height of polygonal shaped structures (with greater than four faces) should be determined from properly scaled wind tunnel tests in accordance with 1.5 of EN 1991-1-4.

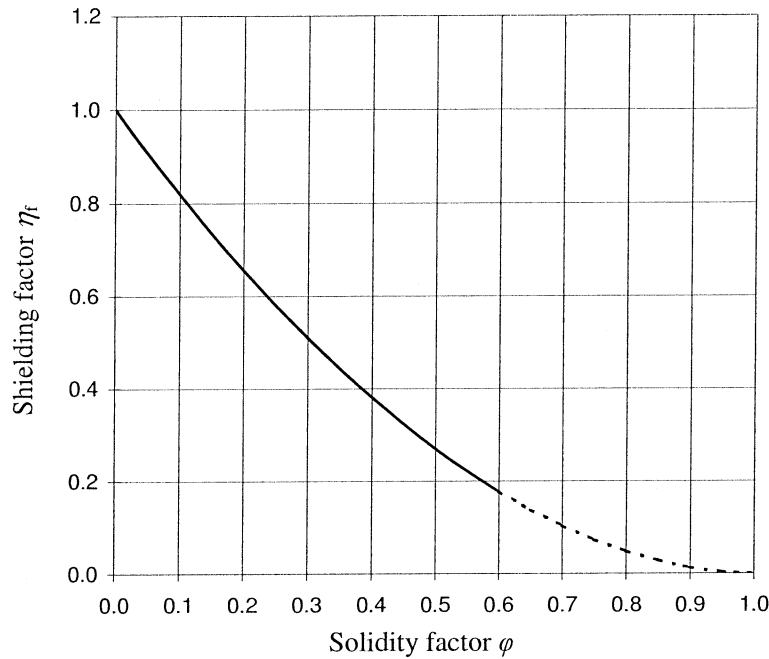


Figure B.2.4 Shielding factor η_f for single frames composed of flat-sided members

B.2.7.2 Wind force coefficients for single frames

- (1) Values of normal force coefficients c_f for single frames composed of both flat-sided and circular-section members should be taken as:

$$c_f = c_{f,f} \frac{A_f}{A_s} + c_{f,c} \frac{A_c}{A_s} + c_{f,c,\text{sup}} \frac{A_{c,\text{sup}}}{A_s} \quad \dots (B.11)$$

where: $c_{f,f}$, $c_{f,c}$ and $c_{f,c,\text{sup}}$ are the normal force coefficients for flat-sided, subcritical circular- and supercritical circular-section members, respectively, given by:

$c_{f,f}$ is the force coefficient for single frames equal to:

$$1,58 + 1,05 (0,6 - \varphi)^{1,8} \text{ for } \varphi \leq 0,6;$$

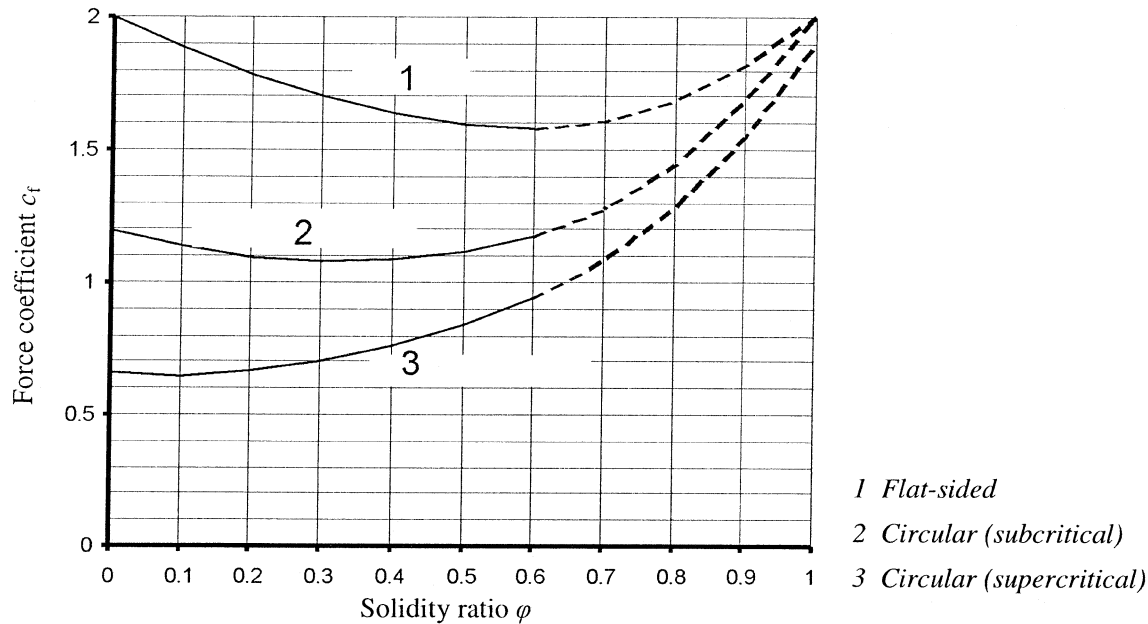
$$1,58 + 2,625 (\varphi - 0,6)^2 \text{ for } \varphi > 0,6;$$

A_f , A_c , $A_{c,\text{sup}}$, A_s and φ are as defined in B.2.7.1.

$$c_{f,c} = (0,6 + 0,4 \varphi^2) c_{f,f}$$

$$c_{f,c,\text{sup}} = (0,33 + 0,62 \varphi^{5/3}) c_{f,f}$$

- (2) Approximate values of these drag coefficients are given in Figure B.2.5.



NOTE: For structures with $\phi > 0,6$ see B.2.7.1(3).

Figure B.2.5 Normal force coefficient c_f for single frames

B.3 Response of lattice towers

B.3.1 Criteria for static methods

(1) The equivalent static method, see B.3.2, should usually be used if the criteria in B.3.1(3) are met. If not, more complex methods such as the spectral analysis method, see B.3.3, should be used. Specialist advice is necessary.

(2) The equivalent static method includes an allowance for the dynamic amplification of response that is typical of the majority of towers likely to be constructed in accordance with this standard. The check for applicability of the static procedure according to equation B.12 should be considered for guidance only. Dynamic augmentation generally increases in successively higher panels of any tower, particularly when supporting large concentrations of ancillary items or when using a concave outline profile (Eiffelization). In such cases caution should be exercised in applying the static procedure to towers where these effects are considerably more than those typically encountered.

(3) The equivalent static procedures may be used if:

$$\frac{7m_T}{\rho_s c_{fT} A_T \sqrt{d_B \tau_o}} \left(\frac{5}{6} - \frac{h_T}{h} \right)^2 < 1 \quad \dots (B.12)$$

where: $c_{fT} A_T$ is the sum of the panel wind forces (including ancillaries), commencing from the top of the tower, such that $c_{fT} A_T$ is just less than one-third of the overall summation $c_f A_T$ for the whole tower (in m^2);

ρ_s is the density of the material of the tower structure (in kg/m^3);

m_T is the total mass of the panels making up c_{fT} (in kg);

h is the height of the tower (in m);

h_T is the total height of the panels making up c_{fT} but not greater than $h/3$ (in m);

τ_o is a volume/resistance constant taken as 0,001 m;

d_B is the depth in the direction of the wind, equal to:

- base d for rectangular towers (in m);
- $0,75 \times$ base width for triangular towers (in m).

B.3.2 Equivalent static method

B.3.2.1 General

(1) For symmetrical towers constructed of leg members with triangulated bracings, with or without ancillaries for which the resistance has been calculated by B.2.2, maximum member forces should be derived in accordance with B.3.2.2.1 to B.3.2.2.5. For unsymmetric towers constructed of leg members with triangulated bracings and containing ancillaries, or for towers for which the resistance has been calculated by B.2.7 the maximum member forces should be determined in accordance with B.3.2.2.6.

NOTE: For symmetric triangular and square towers the wind loads in the cross-wind direction will not govern design and may thus be ignored. For unsymmetric towers these loads are taken into consideration.

B.3.2.2 Wind loading

B.3.2.2.1 General

(1) The wind force in the direction of the wind on the tower should be determined with (5.3) of EN 1991-1-4, but using the wind force coefficients given in B.2 of this Annex.

(2) The mean wind load in the direction of the wind on the tower $F_{m,w}(z)$ should be taken as:

$$F_{m,w}(z) = \frac{q_p}{1 + 7I_v(z_e)} \sum c_f A_{ref} \quad (B14a)$$

(3) The equivalent gust wind load in the direction of the wind on the tower $F_{T,w}(z)$ should be determined from:

$$F_{T,w}(z) = F_{m,w}(z) \left[1 + \left(1 + 0,2(z_m/h)^2 \right) \frac{[1 + 7I_v(z_e)]c_s c_d - 1}{c_o(z_m)} \right] \quad \dots (B14b)$$

where: I_v is the turbulence intensity according to EN 1991-1-4

$c_s c_d$ is the structural factor from section 6.3 of EN 1991-1-4

z_m is the height above the base at which the load effect is required

h is the overall tower height

c_o is the orography factor according to EN 1991-1-4

B.3.2.2.2 Loading for calculating member forces or foundation forces

(1) The maximum member force S_{max} , or forces on foundations should be determined from $F_{m,w}$ and increased by a factor:

$$S_{max} = S_{m,w} \left[1 + \left(1 + 0,2(z_m/h)^2 \right) \frac{[1 + 7I_v(z_e)]c_s c_d - 1}{c_o(z_m)} \right] \quad (\text{see also (B14b)}) \quad \dots (B15)$$

where: S_w is the member force or foundation force determined from the mean wind load $F_{m,w}$.

B.3.2.2.3 Loading for calculating shear forces

(1) The loading to be used to calculate bracing member forces should be based on the configuration of the tower.

- NOTE:** Forces in bracing members at leg slope changes may have components from the leg force and from the shear.

- the mean wind loading, $F_{m,w}(z)$, considered below the intersection and an equivalent ‘gust’ wind load $F_{T,w}(z)$ above the intersection.
- the mean wind loading, $F_{m,w}(z)$, considered above the intersection and an equivalent ‘gust’ wind load $F_{T,w}(z)$ below the intersection.

- NOTE:** For bracing members above the highest intersection point the procedure of B.3.2.2.3(2) may be used.

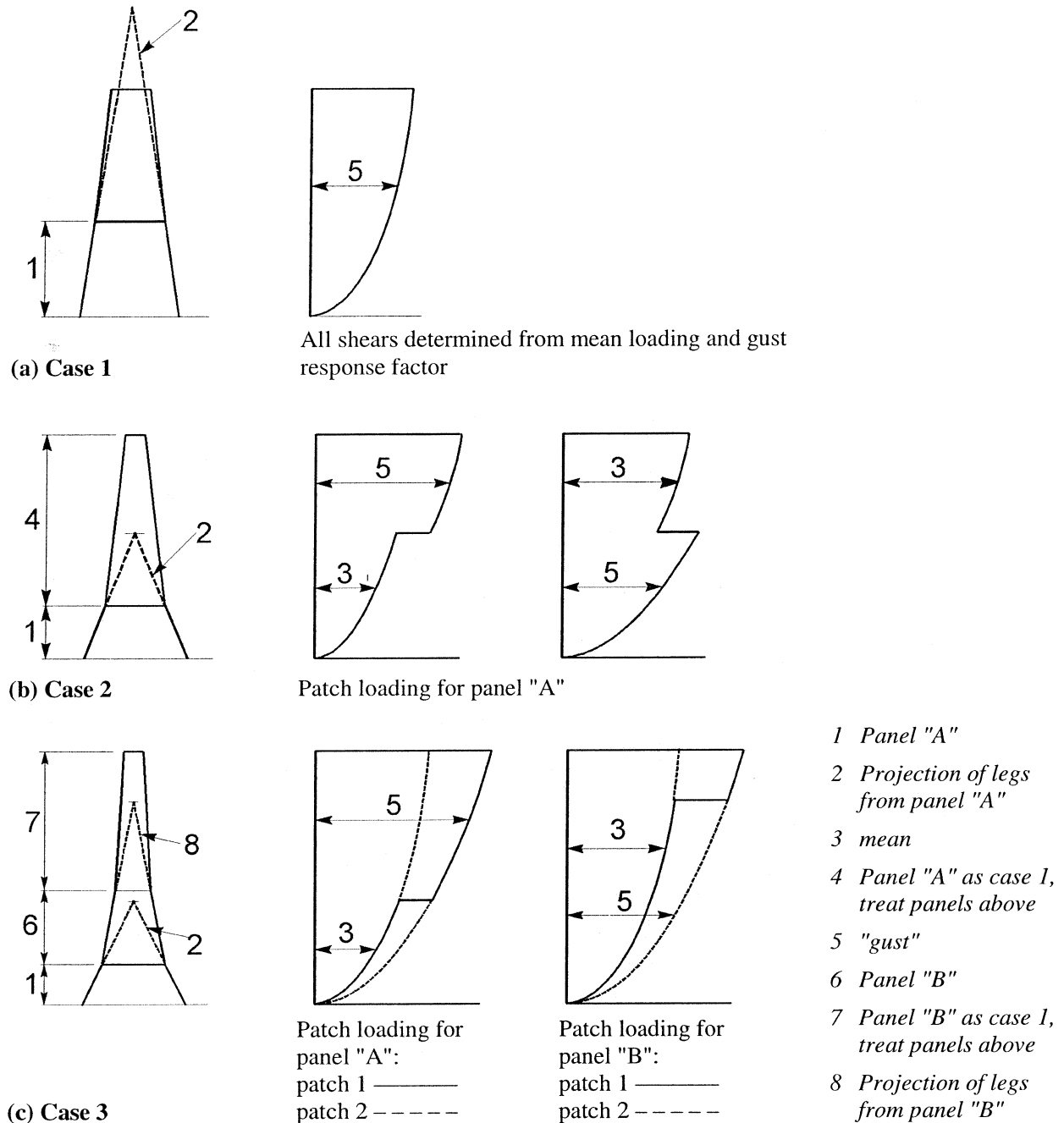


Figure B.3.1 Shear patch loading

B.3.2.2.4 Loading on cables and guys supported by the tower

(1) The maximum wind loading on cables and guys in the direction of wind $F_{c/Gw}(z)$ should be taken as:

$$F_{c/Gw}(z) = \frac{q_p(z)}{1 + 7I_v(z)} \sum c_{f,G} \cdot A_G \cdot \left[1 + \frac{[1 + 7I_v(z)]c_s c_d - 1}{c_o(z)} \right] \quad \dots (B.16)$$

where:

$q_p(z)$ is the peak wind pressure at the effective height of the cable, z metres above site ground level determined in accordance with EN 1991-1-4;

$\sum c_{f,G}$ is the total wind force coefficient on the guy/cable in the direction of the wind, determined in accordance with B.2;

B.3.2.2.5 Loading for calculating deflections and rotations

(1) Deflections and rotations are normally only important to satisfy serviceability requirements. The serviceability criteria should be defined by the client in the project specification (see 7.2.2).

B.3.2.2.6 Wind loading for unsymmetrical towers or towers with complex attachments

(1) For unsymmetrical towers or towers that contain unsymmetrically placed large ancillaries and/or cables imposing significant torsional and crosswind loads, the total forces due to the effect of wind load should allow for the combined action of wind on individual parts, both along wind and crosswind, when appropriate.

(2) The fluctuating load effects caused by cross wind turbulence should be considered in conjunction with along wind load effects.

(3) To determine the total load effects in such cases the mean along wind load effect should be separated from the fluctuating wind load effect. Thus the tower should be analysed under the mean wind load in the direction of the wind ($F_{m,w}(z)$) as determined from B.3.2.2.1(1).

NOTE: If cables are present the mean load on the cables $F_{m,cw}(z)$ should be used (see B.3.2.2.4).

(4) The individual load effects should then be calculated as:

a) the mean wind load effect, $S_{m,TW}$, determined from the mean wind load $F_{m,TW}(z)$.

b) the fluctuating in line wind effect, $S_{1,TW}$, determined from:

$$S_{1,TW} = S_{m,TW} \frac{[1 + 7I_v(z)]c_s c_d - 1}{c_o(z_m)} (1 + 0,2(z_m/h)^2) \quad \dots (B.17)$$

c) Turbulence in the crosswind direction causes fluctuating crosswind load effects ($S_{1,TX}$) which, in the absence of other information should be taken as:

$$S_{1,TX} = K_X \left(\frac{\sum c_x}{\sum c_f} \right) S_{1,TW} \quad \dots (B.18)$$

where: K_X is a factor to allow for crosswind intensity of turbulence.

Σc_X is the crosswind lift coefficient of the structure (and any ancillaries if present) over the panel height concerned.

NOTE 1: The value of K_X may be given in the National Annex. The value $K_X = 1,0$ is recommended.

NOTE 2: Crosswind turbulence will cause fluctuating crosswind loads even in symmetric towers; however such loads will not normally affect the critically loaded elements except for fatigue.

(5) The total load effect ΣS_T in any member due to wind should then be taken as:

$$S_T = S_{m,TW} + S_{m,cw} + \sqrt{S_{1,TW}^2 + S_{1,TX}^2 + S_{cables}^2} \quad \dots (B.20)$$

B.3.3 Spectral analysis method

(1) When response to along wind forces is calculated by a spectral analysis, the meteorological conditions to be assumed should be those defined in EN 1991-1-4, and the wind force coefficients taken as those given in B.2. In addition, the parameters defined in Annex B of EN 1991-1-4 should be adopted in the absence of more accurate information.

NOTE: The National Annex may give further information.

- (2) Cross wind turbulence will cause fluctuating load effects which need to be considered in conjunction with in-line wind loads. Appropriate parameters, consistent with those adopted for downwind effects should be adopted.

NOTE: The National Annex may give further information.

B.3.4 Crosswind vortex vibrations

- (1) If towers support large prismatic, cylindrical or bluff bodies or may be expected to become heavily blocked by icing, their susceptibility to vortex-excited vibrations and/or galloping should be determined, in accordance with EN 1991-1-4.

B.4 Response of guyed masts

B.4.1 General

- (1) The maximum forces to be used in the design of mast components and foundations should be calculated with due allowance for the response to wind turbulence.
- (2) Such forces should represent the resultant effect of an equivalent static loading due to wind of speed equal to the appropriate 10 minute mean value, acting only in the wind direction, and fluctuating loading both downwind and, where relevant, crosswind due to gustiness.

B.4.2 Criteria for static methods

- (1) Generally static analysis procedures can be used to determine the maximum forces in the members of a mast (see B.4.3). Only for masts which may be prone to significant dynamic response is it necessary to undertake dynamic response methods (see B.4.4).
- (2) The design of major masts whose economic consequences of failure or potential hazards resulting from failure are high (see 2.3) should be checked by dynamic response procedures if required by the project specification.
- (3) The following criteria should be satisfied for the static analytical procedures to be used:
- any cantilever has a total length above the top guy level of less than half the spacing between the penultimate and top guys;
 - the parameter β_s is less than 1, where:

$$\beta_s = \frac{4 \left(\frac{E_m I_m}{L_s^2} \right)}{\left(\frac{1}{N} \sum_{i=1}^N K_{Gi} H_{Gi} \right)} < 1 \quad \dots (B.21a)$$

$$\text{with: } K_{Gi} = 0,5 N_i A_{Gi} E_{Gi} \cos^2 \alpha_{Gi} / L_{Gi} \dots \quad (B.21b)$$

where: N the number of guy levels;

A_{Gi} the cross sectional area of guy at level i ;

E_{Gi} the elastic axial modulus for guy at level i ;

L_{Gi} the length of guy at level i ;

N_i the no. of guys attached at level i ;

H_{Gi} the height above the mast base of the i th guy level;

α_{Gi} the slope of the guy chord at level i to the horizontal;

E_m the elastic modulus for the mast;

- I_m the average mast bending inertia;
 L_s the average span between guy levels.

c) The parameter Q is less than 1, where:

$$Q = \frac{1}{30} \sqrt[3]{\frac{HV_H}{D_o}} \sqrt{\frac{m_o}{HR}} \quad \dots (B.21c)$$

- m_o the average mass per unit length of the mast column including ancillaries (kg/m);
 D_o the average face width of the mast (m);
 V_H the mean wind speed V_e at top of mast (m/sec);
 R the average total wind drag obtained from A.2 (m²/m);
 H the height of mast, including cantilever if present (m).

(4) If any of the criteria in (3) are not satisfied, then the spectral analysis method (see B.4.4) should be followed.

B.4.3 Equivalent static methods

B.4.3.1 General

(1) To allow for the dynamic response of masts to wind loading the mast should be analysed for a series of static 'patch' loading patterns based on the mean loading augmented by wind load 'patches'. This procedure requires several static wind analyses for each wind direction considered, the results being combined to provide the maximum response.

(2) For masts of symmetrical structural cross section with triangulated bracing, either without ancillaries or with ancillaries symmetric in the wind direction being considered, and are not likely to be dynamic sensitive (see B.4.7), the maximum forces should be derived in accordance with B.4.3.2.

(3) For masts containing ancillaries which are unsymmetric in the wind direction being considered, the additional forces due to cross wind effects should be determined in accordance with B.4.3.2.8.

B.4.3.2 Load cases to be considered

B.4.3.2.1 Mean wind loading

(1) The wind load in the direction of the wind on the mast column $F_{m,W}$ due to the mean wind should be taken as:

$$F_{m,W} = \frac{q_p(z)}{1 + 7I_v(z)} \sum c_w(z) A \quad \dots (B.22)$$

where: $c_w(z)$ is the wind force coefficient of the structure (and any ancillaries if present) in the direction of the wind over the mast section concerned, at a height z metres above the site ground level, determined in accordance with B.4.2

(2) The loads should be taken as acting at the level of the centre of areas of faces (including ancillaries if present) within the section height.

(3) The wind loading on the guys, F_{GW} , normal to the guys in the plane containing the guy and the wind, due to the mean wind should be taken as:

$$F_{GW} = \frac{q_p(z)}{1 + 7I_v(z)} c_G(z) A \quad \dots (B.23)$$

where: $c_G(z)$ is the wind force coefficient of the guy under consideration determined in accordance with B.2;

(4) If a uniform loading is used then $q_p(z)$ should be taken as the wind speed at $\frac{2}{3}$ the height of the relevant guy attachment to the mast.

(5) The load effects S_m due to the mean wind should be determined for each component of the mast by a geometric non-linear static analysis under the mean loading $F_{m,w}$ and F_{GW} .

B.4.3.2.2 Patch loads

(1) In addition to the mean loading derived from B.4.3.2.1 successive patch loads should be applied as follows:

- on each span of the mast column between adjacent guy levels (and the span between the mast base and the first stay level);
- over the cantilever if relevant;
- from midpoint to mid point of adjacent 'spans';
- from the base to the mid height of the first guy level;
- from the mid height of the span between the penultimate and top guy if no cantilever is present, but to include the cantilever if relevant.

(2) These are shown in Figure B.4.1. The 'patch' load should be taken as:

$$F_{PW} = 2k_s \frac{q_p(z)}{1 + 7I_v(z)} \frac{I_v(z)}{c_0(z)} \sum c_w(z) A \quad \dots (B.24)$$

where: $c_w(z)$ is as defined in B.4.3.2.1;

k_s is a scaling factor;

$I_v(z)$ is the turbulence intensity as given in 4.4 of EN 1991-1-4, depending on the site terrain and the orography.

$c_0(z)$ is the orography factor determined from EN 1991-1-4.

NOTE 1: The scaling factor k_s accounts for the multi-modal response of guyed masts.

NOTE 2: The value of k_s may be given in the National Annex. The value $k_s = 3,5$ is recommended.

NOTE 3: For simplicity uniform patch loads may be used taking z as the height at the top of the patch for $I_v(z)$ and $q_p(z)$.

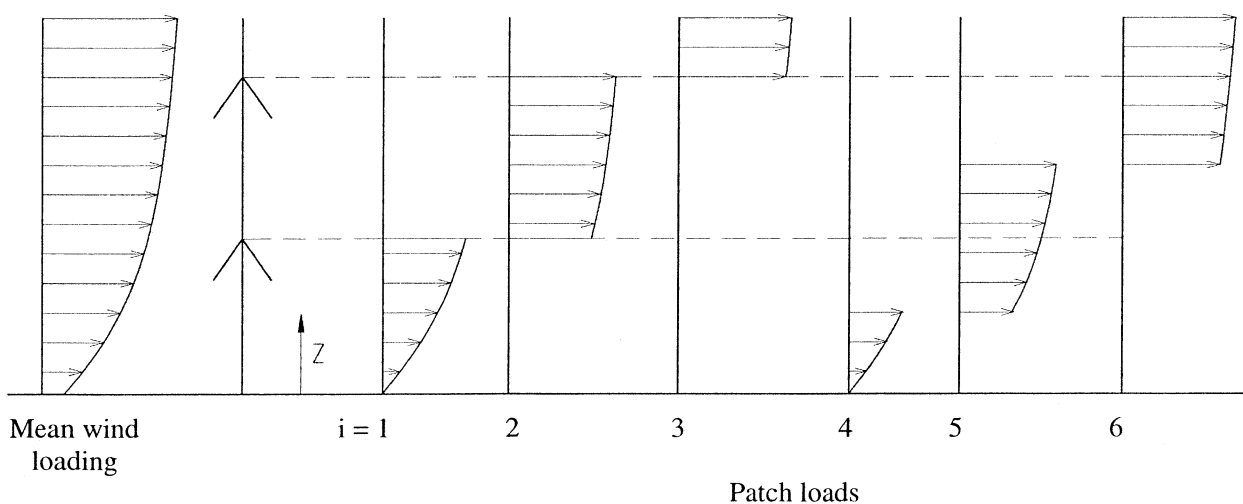


Figure B.4.1 Application of patch loads

(3) These patch loads should be applied to the mast, under mean wind loading determined from B.4.3.2.1.

(4) For masts up to 50m height only one case needs to be considered, with the mean and patch load enveloping the mast.

NOTE 1: In such cases the shear bracing in each span should be designed for the maximum shear (and associated torsion) in that span.

NOTE 2: In such cases the legs and their connections in each span should be designed for the maximum (and minimum) leg load in that span.

NOTE 3: In such cases if the mast supports a cantilever, then (i) mean plus patch loading on the cantilever and mean load on the mast and (ii) mean load on the cantilever and mean plus patch loading on the mast should also be considered.

B.4.3.2.3 Loading on guys

(1) For each patch loading case on the mast column, as given in B.4.3.2.2 patch wind loads, F_{PG} , should be applied within the same boundaries, see Figure B.4.2. These patch loads should be applied normal to each guy in the plane containing the guy and the wind, and taken as:

$$F_{PG} = 2k_s \frac{q_p(z)}{1 + 7I_v(z)} \frac{I_v(z)}{c_0(z)} c_G(z) A \quad \dots (B.25)$$

where: k_s is a scaling factor;

$c_G(z)$ is the wind force coefficient normal to the guy in the plane containing the guy and the wind determined in accordance with B.2.

NOTE 1: The scaling factor k_s accounts for the multi-modal response of guyed masts.

NOTE 2: The value of k_s may be given in the National Annex. The value $k_s = 3,5$ is recommended.

(2) For simplification the patch loading may be ‘smeared’ over the whole height of the guys in question by multiplying the above wind load by the ratio z_p/z_G :

where: z_p is the “height” of the patch on the actual guy; and

z_G is the height to the attachment of the guy to the mast.

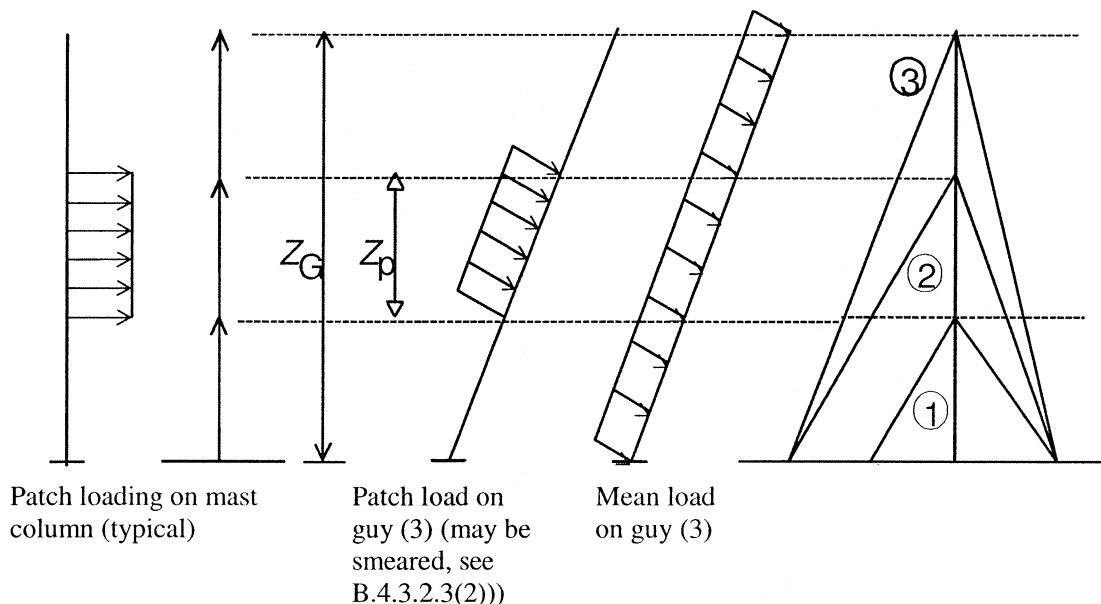


Figure B.4.2 Patch loading on guys

B.4.3.2.4 Derivation of response under patch loads

- (1) The load effect in each element of the mast column and guys derived from each patch load applied successively, S_{PLi} , should be calculated.
- (2) This should be done by calculating the difference between the load effect from the patch load combined with the mean load and the load effect of the mean load alone.
- (3) These load effects should then be combined as the root sum of squares, or:

$$S_p = \sqrt{\sum_{i=1}^N S_{PLi}^2} \quad \dots (B.26)$$

where: S_{PLi} is the load effect (response) from the i th load pattern;

N is the total number of load patterns required;

S_p is the total effective load effect of the patch loads.

B.4.3.2.5 Total load effects

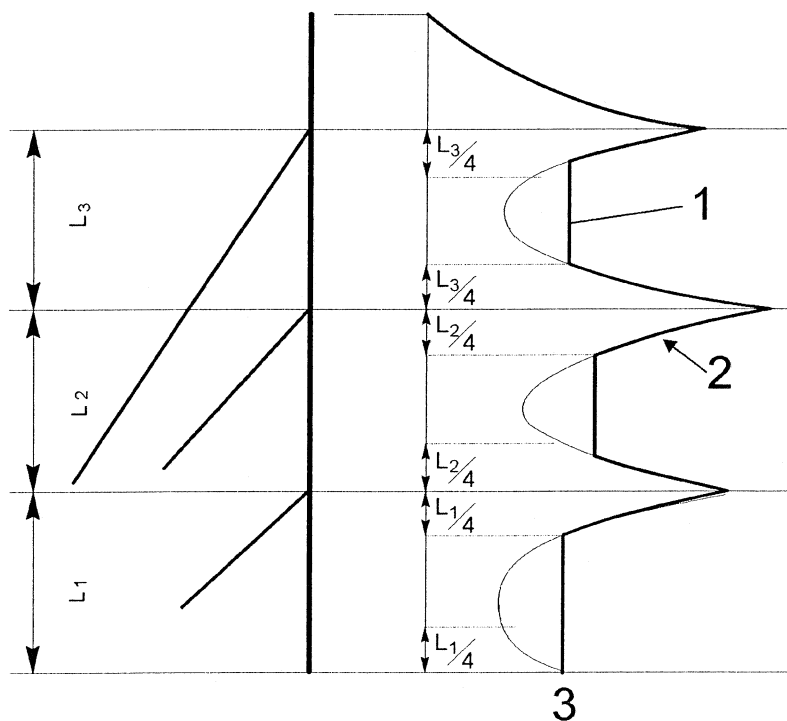
- (1) The total load effects for each component of the mast column, S_{TM} , should be determined from:

$$S_{TM} = S_M \pm S_p \quad \dots (B.27)$$

where: S_M is the mean load effect determined from B.4.3.2.1;

S_p is the fluctuating load effect determined from B.4.3.2.4 using the sign to produce the most severe effect.

- (2) In the calculation of the total force in the shear bracing in each span of the mast column in accordance with (1) above, the minimum value within that span should be taken as the highest calculated at a distance of one quarter of the span from either adjacent guy attachment levels (or the mast base if relevant). In this context 'span' refers to the distance between adjacent guy levels or between the base and the lowest guy level. (See Figure B.4.3.)



1 Minimum value to be used in this span

2 See NOTE

3 Force in shear bracing

NOTE: Envelope of forces in bracing members arising from patch loading (absolute values shown)

Figure B.4.3 Minimum forces in shear bracing in mast column

B.4.3.2.6 Wind directions to be considered

- (1) For each member of the mast, the wind direction giving the most severe total load effect should be considered. This in practice means that several wind directions should be investigated.
- (2) If the mast is nearly symmetrical in geometry and loading, at least three wind directions should be analysed for a triangular mast guyed in three directions, i.e. 90° , 30° to a face and 60° to a face. For a mast with square cross section and guyed in four directions, at least two wind directions should be analysed, normal to a face and 45° to a face. Examples are shown in Figure B.4.4.

NOTE: To account for overall buckling of symmetric masts (see 5.1(5)) introduction of a lateral effect (such as a cross-wind force of 2% of the along wind force or a wind direction of 2° off the notional wind direction) should be provided in undertaking the second order analysis.

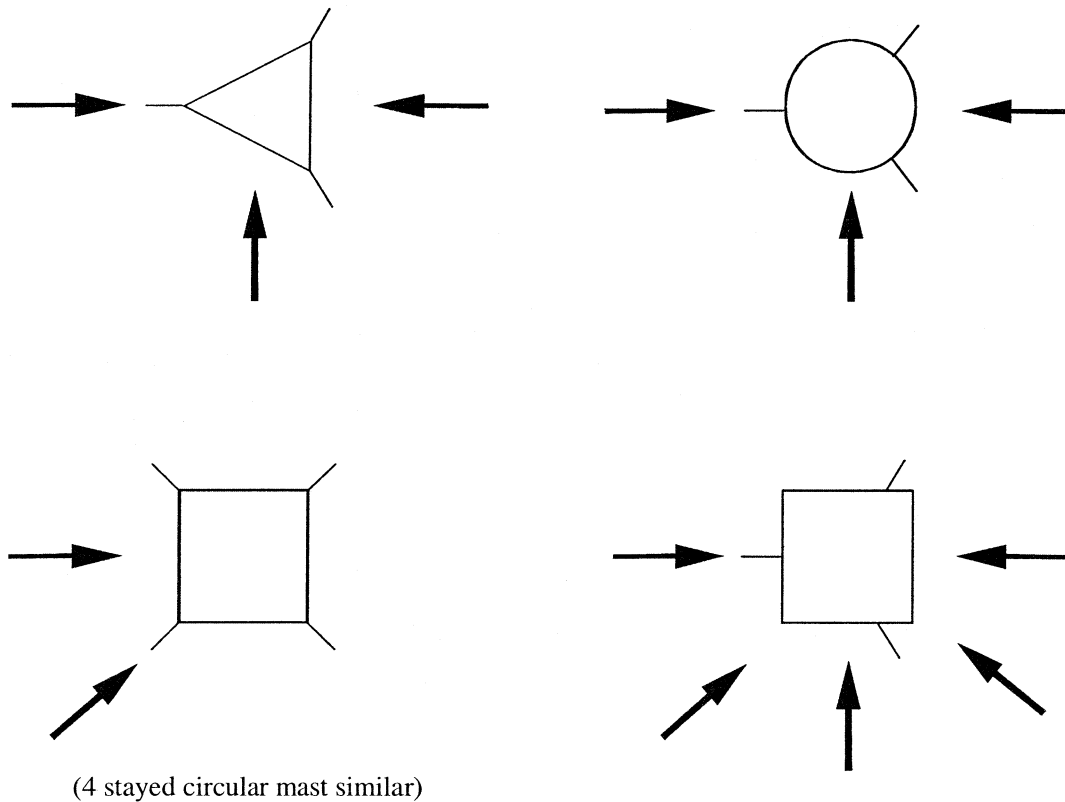


Figure B.4.4 Typical wind directions to be considered

B.4.3.2.7 Loading for calculating deflections and rotations

- (1) Deflections and rotations are normally only important to satisfy serviceability requirements. The serviceability criteria should be defined by the client in the project specification (see 7.2.2)

B.4.3.2.8 Wind loading for unsymmetrical masts or masts with complex attachments

B.4.3.2.8.1 General

- (1) For unsymmetrical masts or masts that contain unsymmetrically placed large ancillaries and/or cables imposing torsional and cross wind loads, the total forces due to the effects of wind load should allow for the combined action of wind on individual parts, both along wind and crosswind, when appropriate.
- (2) Cross wind turbulence will cause fluctuating load effects. This may need to be considered in conjunction with along wind loads.
- (3) The procedure for separating the mean along wind loads from the fluctuating loads needs to be carried out, as set out for towers in B.3.2.2.1. For guyed masts, however this will necessitate a series of transverse patch wind loads to be applied in a similar manner to those for along wind as set out in B.4.3.2.2.
- (4) The total load effects should then be determined from:

$$S_{TM} = S_M \pm \sqrt{S_{PW}^2 + K_X^2 S_{PX}^2} \quad \dots (B.28)$$

where: S_{PW} is the load effect from the in-line patch loads;

S_{PX} is the load effect from the cross-wind patch loads;

K_X is a factor to allow for cross wind intensity of turbulence

NOTE 1: The value of K_x may be given in the National Annex. The value $K_x = 1,0$ is recommended.

NOTE 2: Cross wind turbulence will cause fluctuating cross wind loads even in symmetric masts; however such loads will not affect the critically loaded elements.

(5) Alternatively, for simplification the cross wind turbulence effects need not be calculated explicitly as in B.4.3.2.8.1(4) above but the in-line peak load effects, S_{TM} , from B.4.3.2.5(1) should be increased by 10% to allow for cross wind effects.

B.4.4 Spectral analysis method

(1) When response is calculated by spectral analysis this should be used for the resonance contribution to the response only.

(2) The non-resonant response may be determined using the general static procedure (See B.4.3.2) using $k_s = 3,5$.

(3) The meteorological conditions to be assumed should be those defined in EN 1991-1-4, and the wind resistance taken as that given in B.2. In addition, the parameters defined in Annex B of EN 1991-1-4 should be adopted in the absence of more accurate information.

(4) Cross wind turbulence will cause fluctuating load effects which need to be considered in conjunction with along wind loads. Appropriate parameters, consistent with those adopted for along wind effects should be adopted.

(5) Response should be calculated for all modes of vibration having natural frequencies less than 2 Hz.

B.4.5 Vortex-excited vibrations

(1) When masts support large bluff bodies or are likely to become heavily blocked by icing, then susceptibility to vortex-excited vibrations, should be taken into account in accordance with EN 1991-1-4.

B.4.6 Guy vibrations

(1) The mast guys should be checked for high frequency vortex-excited vibrations and guy galloping, particularly when the guys are iced, as follows:

a) *Vortex excitation*

Guys may be subject to low amplitude resonant type vibrations at low wind speeds caused by vortex excitation at high frequency.

As excitation can occur in high modes general rules cannot be set down. However as a guide, experience shows that such vibrations are likely to occur if the still air tensions in the guys are in excess of ten per cent of their breaking load.

b) *Galloping (including rain induced vibrations)*

Guys may be subject to galloping excitation when coated with ice or thick grease. The accretion of ice or grease can form aerodynamic shapes which provide lift and drag instabilities. These result in low frequency high amplitude vibrations. Similar vibrations are also known to occur under conditions of rain.

Again general rules cannot be provided as the occurrence of galloping is critically dependent on the formation of ice, or profile of grease. It will generally only occur on large diameter guys and is relatively insensitive to initial stay tensions. See EN 1993-1-11 (Clause 8.3)

(2) If guy vibrations are observed, dampers or spoilers should be provided as required to limit the resulting stresses, see D.2.

(3) Fatigue checks of the anchorages should be made if such vibrations are known to have occurred and no remedial action has been taken. In such cases specialist advice should be sought.

Annex C [informative] – Ice loading and combinations of ice with wind

NOTE: As this Annex deals with ice loading and combinations of ice with wind for masts and towers it is expected that it will be transferred to EN 1991 - Actions on structures.

C.1 General

- (1) Atmospheric ice loading on masts and towers can, for exposed sites, grow to considerable thicknesses, and combined with wind the increased wind drag due to iced members might in some instances govern the design.
- (2) The magnitude of ice deposit on structures, as well as the density, the placing and the shape of the ice on masts and towers heavily depends on the local meteorological conditions and the topography and the shape of the structure itself.
- (3) Atmospheric icing is traditionally classified according to two different formation processes:
 - in-cloud icing;
 - precipitation icing.
- (4) These may result in various types of ice as soft rime, hard rime, wet snow and glaze, having different physical properties concerning density, adhesion, cohesion, colour and shape. For instance the density varies typically from approximately 200 kg/m^3 to 900 kg/m^3 , and from a concentric deposit (glaze and wet snow) to an eccentric deposit on one face pointing windward for soft and hard rime.
- (5) For engineering design purposes it is traditionally assumed that all members of a mast or a tower are covered with a certain ice thickness, which together with a density may be used for calculation of the weight of the ice as well as the wind drag. Such methods may be justified in areas where glaze or wet snow form the design ice load, but in the case of rime the physical reality does not coincide with a uniform ice thickness on all members of towers and masts. However in areas where the ice deposit from in-cloud rime is relatively small, the method of calculating ice weight and wind drag with ice assuming a uniform ice cover can be practical and reasonable if conservative values are used.
- (6) On the other hand there are areas in Europe that are very exposed to heavy atmospheric icing and for such areas the ice load should be estimated by experts in atmospheric icing. This should include the weight, the location, the shape, etc. of the ice load on the actual structure, as well as the appropriate combination of ice with wind which should be specified in detail.
- (7) The following clauses provide a general description of how to treat ice load and ice in combination with wind on towers and masts.

C.2 Ice loading

- (1) The principles for characteristic ice loading inclusive of the density and other design parameters is given in ISO 12494. In ISO 12494 the ice load is based on Ice Classes for rime and glaze, but the actual Ice Class for the location is not given, nor is the density of the ice.

NOTE: The National Annex may give further information.

- (2) As the ice may deposit asymmetrically on towers and masts, such situations should be taken into account. Asymmetrical icing is of particular interest for masts where icing on the different guys may vary considerably causing bending effects in the mast column. Asymmetrical ice on the guys may partly be caused by asymmetrical ice accretion depending on wind direction and partly caused by unequal shedding of ice from the guys.

C.3 Ice weight

(1) When estimating the weight of the ice on a lattice tower or mast column, it may normally be assumed that all structural members, components of ladders, ancillaries, etc. are covered with ice having the same thickness over the whole surface of the member, see Figure C.1.

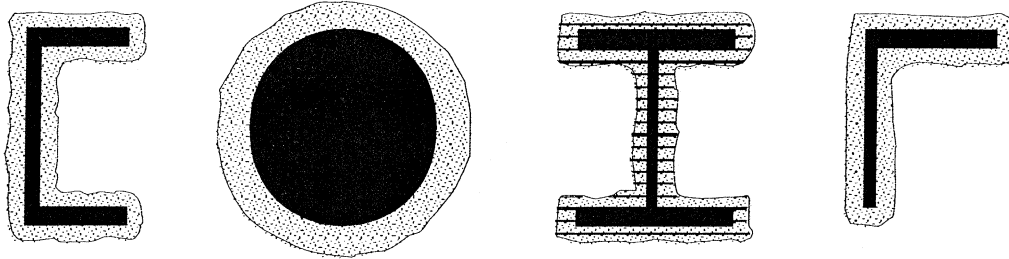


Figure C.1 Ice thickness on structural members

C.4 Wind and ice

(1) In areas where atmospheric icing occurs, combinations with wind can often govern the design of masts and towers. The increased wind drag caused by the ice deposit on the individual members might thus result in critical loading, even though the associated wind speeds are less than the maximum characteristic values.

(2) The wind drag of an iced tower or mast may for glaze ice be estimated using the same basic procedure as given in Annex B, taking into account the increased width of members and component due to the ice thickness. If the gaps between elements are small (say less than 75mm) they may be assumed to be closed by ice. For rime ice estimation of the wind drag is far more complicated, and for fully iced mast or mast faces special attention should be taken. Guidelines are given in ISO 12494.

(3) When combining ice and wind load, the characteristic wind pressure in periods where atmospheric icing can occur is less than the characteristic wind pressure in all situations. This may be taken into account by multiplying the characteristic wind pressures in EN 1991-1-4 by a factor k . The factor k is given in ISO 12494 dependent of the Ice Class.

C.5 Asymmetric ice load

(1) Asymmetric icing on a mast should be taken into consideration by applying the appropriate ice to the mast shaft and to all guys apart from:

- one lane of the top guy level; and as a separate case;
- two lanes of the top guy level.

C.6 Combinations of ice and wind

(1) Two combinations of wind and ice should be taken into consideration for both symmetrical icing and asymmetrical icing. The design values of the loads are as given in 2.3 and the following combinations should be used:

- for dominant ice and accompanying wind:

$$\gamma_G G_k + \gamma_{ice} Q_{k,ice} + \gamma_W k \psi_W Q_{k,w} \quad \dots (C.1)$$

- for dominant wind and accompanying ice:

$$\gamma_G G_k + \gamma_W k Q_{k,w} + \gamma_{ice} \psi_{ice} Q_{k,ice} \quad \dots (C.2)$$

where k is defined in C.4(3).

NOTE: The National Annex may give information on combination factors. The following combination factors are recommended:

$$\psi_W = 0,5 \quad \dots (C.3a)$$

$$\psi_{ice} = 0,5 \quad \dots (C.3b)$$

- (3) For partial factors on dead load γ_G , ice load γ_{ice} and wind load γ_W see Annex A.

2

2

2

2

Annex D [normative] – Guys, dampers, insulators, ancillaries and other items

D.1 Guys

D.1.1 Metallic guys and tension elements

- (1) For metallic guys and tension elements see EN 1993-1-11.
- (2) Filling material in antennas should be metallic

NOTE: The National Annex may give further information.

D.1.2 Non metallic guys

- (1) Materials other than steel may be used provided that they have an acceptable modulus of elasticity and provided that appropriate measures are taken to prevent vibrations in higher frequencies.

NOTE: In the selection of Synthetic materials the low modulus of elasticity of some products may require a higher initial tension to compensate for their lower stiffness, which can lead to possible high frequency vibrations. The ends of such ropes are sealed to prevent entrance of moisture which might otherwise lead to the discharge of lightning. Partial factors for non metallic guys may need to be higher than for steel guys.

- (2) Non metallic guys should comply with the relevant technical specification.

NOTE: The National Annex may give further information.

D.2 Dampers

D.2.1 Structure dampers

- (1) The possible structural vibrations that can occur in a tower or mast under wind should be reduced, if necessary, by the use of the damping devices.

NOTE: See Annex B and Annex A of EN 1993-3-2.

D.2.2 Guy dampers

D.2.2.1 General

- (1) To suppress the possible vibrations that can occur in guys under wind one of the following procedures should be followed:
 - a) Dampers should be mounted on guys in all cases where the initial tension is greater than 10% of the rated breaking strength of the guy.
 - b) Where guy dampers are not fitted the guys should be carefully observed during the first years of service to ensure that excessive frequency and/or amplitude of oscillations are not occurring. Otherwise dampers as described in (a) should be fitted.

NOTE: For vibrations see Annex B.

D.2.2.2 Dampers to reduce vortex excitation

- (1) Appropriate dampers should be installed in all cases where unacceptable vortex-excited vibrations are predicted or have been observed. Dampers should conform to appropriate technical specifications. A frequency band of vibration should be specified.

D.2.2.3 Dampers to prevent galloping (including rain/wind induced vibrations)

- (1) Partial control of galloping and rain/wind induced vibrations may be obtained by the attachment of a rope from guy to guy, connecting the points of maximum amplitude of two or more guys. The effect of this under high wind conditions should be taken into account in the design of the connections to the guy.

NOTE: Hanging chains may also be used to provide partial control of galloping, if the chains will operate over the relevant frequency range.

D.3 Insulators

- (1) Insulators should be selected dependent on electrical and mechanical requirements.
- (2) The minimum ultimate strength should be taken from relevant technical specifications.
- (3) Each guy insulator fitting should be designed such that even if an insulator suffers electrical failure the stability of the mast is still ensured. This may be achieved, for example by the use of fail safe insulators or insulators in parallel.
- (4) Arcing arrangements should be made such that arcing will not occur along the surface of the insulating materials adjacent to the steel fitting.
- (5) Where insulators are used at the base of the mast, jacking facilities should be provided to enable replacement of units.
- (6) Mechanical loading and unloading for ceramic insulating material (during mechanical tests and/or during construction) should be carried out in accordance with the relevant technical specifications.

NOTE 1: The National Annex may give further information. In the absence of other data loading and unloading should be undertaken at a rate of approximately 5% of the expected load in steps of approximately 1 minute, such that any loading or unloading will take not less than 20 minutes.

NOTE 2: For electrical properties see National Annex.

D.4 Ancillaries and other items

D.4.1 Ladders, platforms, etc.

- (1) Ladders, platforms, safety rails and other ancillaries should comply with the relevant specifications.

NOTE: The National Annex may give further information.

D.4.2 Lightning protection

- (1) Towers, masts and guys should be effectively earthed for protection against lightning. This may be achieved by a metallic tape ring around the base connected to metallic plates and rods embedded in the soil. Guy anchors should be similarly protected.
- (2) The earthing system should be completed before erection of the steelwork, and connections should be made to the stay earthing system as erection work proceeds.
- (3) Provided that all the structural joints are electrically continuous, no further bonding need be incorporated.

NOTE: The National Annex may give further information.

D.4.3 Aircraft warning

EN 1993-3-1:2006 (E)

- (1) Structures that constitute a hazard to aerial navigation should be marked.

NOTE: The National Annex may give further information.

D.4.4 Protection against vandalism

- (1) Suitable protective measures should be installed to restrict access by unauthorized persons.

NOTE: The National Annex may give further information.

Annex E [informative] – Guy rupture

E.1 Introduction

- (1) Guy rupture is an accidental action. For partial factors see Annex A.
- (2) The precise analysis of a guyed mast for the dynamic actions caused by a sudden rupture of a guy is very complicated, because several of the different factors influencing the behaviour of the mast immediately after failure are uncertain; for instance the character of the rupture, the damping of the guys and the mast, the vibration of the guys and the mast, etc. Accordingly, the simplified analytical model given in E.2 may be adopted. A conservative procedure is given in E.3.

E.2 Simplified analytical model

- (1) For the simplified analysis of a guyed mast due to the rupture of a guy, the dynamic forces should be assumed to be equivalent to a static force acting on the mast at the level of the set of guys where rupture has assumed to have occurred.
- (2) For the calculation of this static equivalent force $F_{h,dyn,Sd}$ described below, it is assumed that:
- the rupture is a simple cut through the guy;
 - the elastic energy stored in guy 1 (see Figure E.1) before the rupture occurs is neglected;
 - damping is not taken into account;
 - the wind loading when calculating the equivalent force is neglected.

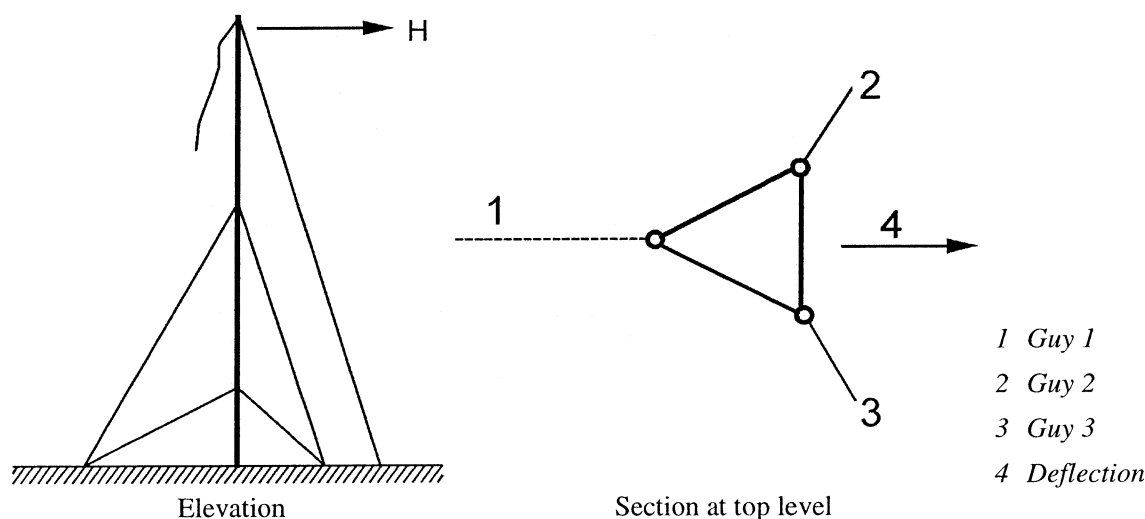


Figure E.1 Guy rupture

- (3) For a given deflection u guys 2 and 3 act on the mast shaft with a force $F_{h,Sd}$. The relation is shown in Figure E.2 as curve 1. It will be seen that $F_{h,Sd}$ decreases with increasing deflection owing to slackening of the guys.
- (4) For the mast system, except for the set of guys at the considered level, the relation between an external horizontal force and the deflection of the node can be shown as well. In Figure E.2 this relation is shown in curve 2. Where the two curves 1 and 2 intersect, the two forces are equal, i.e. there is static equilibrium. The force acting on the joint is $F_{h,stat,Sd}$.
- (5) At the moment that rupture occurs, energy is stored in guys 2 and 3. When the mast starts deflecting, this energy will partially be transformed into kinetic energy.

(6) At the maximum deflection, the kinetic energy will be zero, because the energy lost in guys 2 and 3 has been transferred to the mast as elastic strain energy in the shaft and the guys. Damping has not been taken into consideration.

(7) The energy lost in guys 2 and 3 should be assumed to be equal to the area A2 below curve 1 in Figure E.2.

(8) The deflection resulting in the two areas A1 and A2 being equal, should be taken as the dynamic deflection u_{dyn} .

(9) The dynamic force $F_{h,dyn,Sd}$ corresponds to this dynamic deflection. The impact factor Φ may be determined using:

$$\Phi = \frac{F_{h,dyn,Sd}}{F_{h,stat,Sd}} \quad \dots (E.1)$$

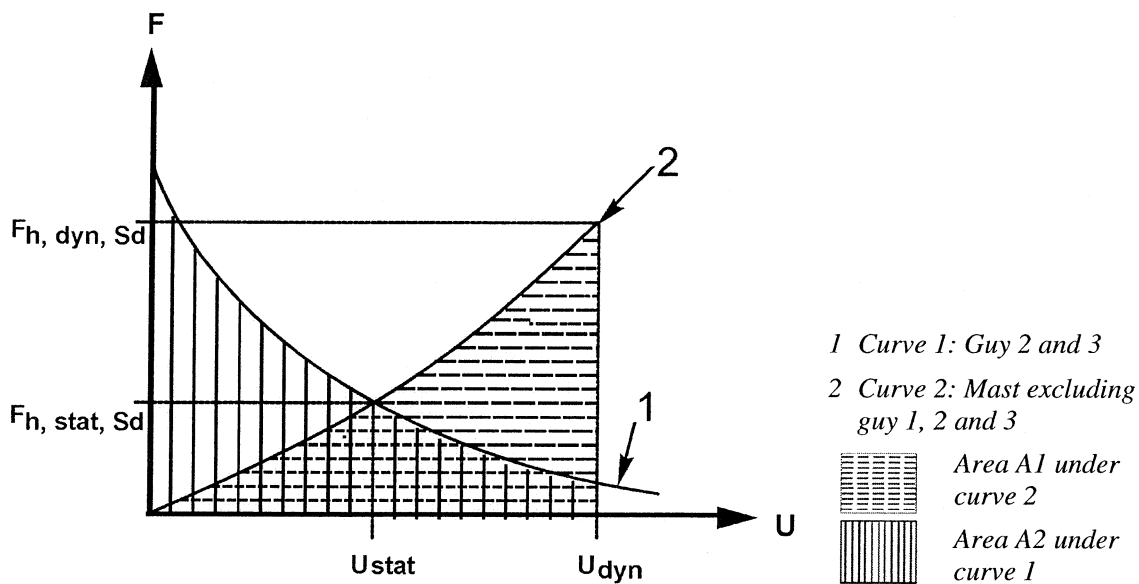


Figure E.2 Force-deflection diagram

(10) The above procedure for the analysis of a mast just after a possible guy rupture has occurred applies to a mast guyed in 3 directions. For masts guyed in 4 (or more) directions similar procedures based on the same principles should be adopted.

(11) If agreed between the designer, the client and the competent authority the above dynamic force arising from the rupture should not be combined with climatic loads.

E.3 Conservative procedure

(1) The dynamic forces in the mast column and the guys caused by a cable rupture can be conservatively estimated using the following static calculations.

(2) The horizontal component of the guy force acting in the guy before the rupture should be used as an additional force acting on the mast without the broken guy.

NOTE: In the absence of other climatic loads this corresponds to the initial tension.

(3) The resulting guy forces should be increased by the factor 1.3 in the case of masts with 2 stay levels or if the rupture of a top guy is considered.

E.4 Analysis after a guy rupture

- (1) In addition to the procedures set out in E.2 or E.3 above, immediately after the rupture of a guy the mast should be able to withstand wind loads for a short period until temporary guying can be arranged.
- (2) If no other requirements are given, the mast without the ruptured guy should be able to withstand a reduced wind load, acting as a static load, and without patch wind loading. The reduced wind loading should be taken as 50% of the characteristic mean wind loading, acting in the most adverse wind direction.

Annex F [informative] – Execution

F.1 General

- (1) Towers and masts should be fabricated and erected according to EN 1090-2.

F.2 Bolted connections

- (1) All bolt assemblies on towers and masts should be provided with suitable measures to avoid any loosening of nuts in service.
- (2) Bolt holes in members should be drilled, where fatigue effects cannot be ignored.
- (3) Fitted or friction grip bolts, or closer tolerances on bolt holes than those given in EN 1993-1-1 may be used where displacements are critical (see 4.2.2).

F.3 Welded connections

- (1) The quality of welds assumed in selecting the appropriate fatigue class of a structural detail, see 9.3, should be specified on the drawings for the fabrication of the structure.

F.4 Tolerances

F.4.1 General

- (1) The tolerances given in EN 1090, Part 2 should be satisfied in fabrication.
- (2) Tighter tolerances should be used where tolerances from EN 1090-2 do not satisfy the requirements for the function of the structure.

F.4.2 Erection tolerances

F.4.2.1 Lattice towers

- (1) The maximum displacement of the tower top should be specified.

NOTE: The National Annex may give further information. A maximum displacement of the tower top not more than $\frac{1}{500}$ of the height of the tower is recommended.

- (2) Final plumbing should be done in calm conditions taking due account of any temperature effects.

F.4.2.2 Guyed masts

- (1) The sensitivity to the structure to varying wind speeds for final plumbing and guy tensioning should be determined in design.

NOTE: Generally if such operations are to be undertaken in wind speed in excess of 5m/s calculations will be required to compensate for the effects of wind, taking due account of any temperature effects.

- (2) Final plumbing and tensioning of guys should normally proceed from the lowest guy level upward.

NOTE: The National Annex may give limits for the tolerances. The following values are recommended:

- a) The final position of the centre line of the mast should all lie within a vertical cone with its apex at the mast base and with a radius of $1/1500$ of the height above the mast base. This does not apply to halyards or aerial array wires.
- b) The resultant horizontal component of the initial guy tensions of all the guys at a given level should not exceed 5% of the average horizontal component of the initial guy tension for that level. The initial tension in any individual guy at a given level should in no case vary more than 10% from the design value, see EN 1993-1-11.
- c) Maximum initial deflection of the mast column between two guy levels, where L is the distance between the guy levels in question, should be $L/1000$.
- d) After erection the tolerance on the alignment of 3 consecutive guy connections on the shaft is limited to $(L_1 + L_2)/2000$, where L_1 and L_2 are the lengths of the two consecutive spans of the shaft.

F.4.3 Tensioning constraints

- (1) After erection, the guys should be tensioned in accordance with the design calculations, taking due account of the actual temperature on the site, see EN 1993-1-11.
- (2) In order to minimise the possibility of guy vibrations still air tensions should be selected such that for each guy the tension is less than 10% of its breaking load.

NOTE 1: For small masts this figure may be exceeded.

NOTE 2: Low still air tensions can give rise to galloping of guys.

F.5 Prestretching of guys

- (1) In order to ensure that the rope is in a truly elastic condition guys should be prestretched preferably prior to terminating. This may be done at the supplier's works or, if suitable facilities exist, at the erection site, see EN 1993-1-11.

NOTE: The need for prestretching is dependent on the planned programme for retensioning, the type and size of the rope used and the sensitivity to deflections.

- (2) Prestretching should be carried out by loading the guy cyclically between 10% and 50% of its breaking load. The number of cycles should not be less than ten. This process should not be carried out by passing the loaded guy around a sheave wheel.

Annex G [informative] – Buckling of components of masts and towers

G.1 Buckling resistance of compression members

(1) The design buckling resistance of a compression member in a lattice tower or mast should be determined according to EN 1993-1-1 as:

$$N_{b,Rd} = \frac{\chi A f_y}{\gamma_{M1}} \text{ for class 1,2 and 3 cross section} \quad \dots (G.1a)$$

$$N_{b,Rd} = \frac{\chi A_{eff} f_y}{\gamma_{M1}} \text{ for class 4 cross sections} \quad \dots (G.1b)$$

where χ is the reduction factor for the relevant buckling mode defined in 6.3.1.2 of EN 1993-1-1.

(2) For constant axial compression in members of constant cross section, the reduction factor χ and the factor ϕ to determine χ should both be determined with the effective slenderness ratio $\bar{\lambda}_{eff}$ instead of $\bar{\lambda}$. The effective slenderness ratio $\bar{\lambda}_{eff}$ is defined as:

$$\bar{\lambda}_{eff} = k \bar{\lambda} \quad \dots (G.2)$$

where

k is the effective slenderness factor obtained from G.2 and

$$\bar{\lambda} = \frac{\lambda}{\lambda_1}$$

λ_1 is defined in EN 1993-1-1

λ is the slenderness for the relevant buckling mode see Annex H.

NOTE: The effective slenderness takes into account the support conditions of the compression member.

(3) For single angle members which are not connected rigidly at both ends (at least with two bolts, if bolted), the design buckling resistance defined in G.1(1) should be reduced by a reduction factor η .

NOTE: The reduction factor η may be defined in the National Annex. The following values are recommended:

$\eta = 0,8$ for single angle members connected by one bolt at each end;

$\eta = 0,9$ for single angle members connected by one bolt at one end and continuous or rigidly connected at the other end.

G.2 Effective slenderness factor k

(1) In order to calculate the appropriate generalised slenderness of the member, the effective slenderness factor k may be determined according to the structural configurations.

(a) *Leg members*

k should be obtained from table G.1.

(b) *Diagonal bracing members*






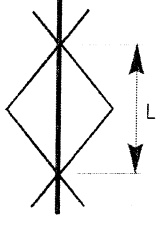
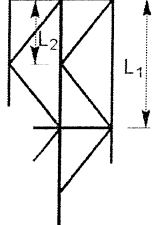
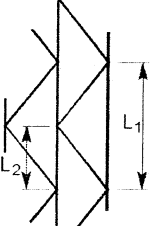
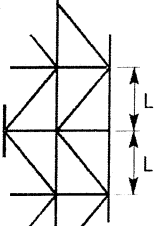
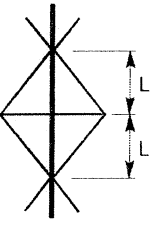
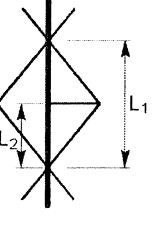
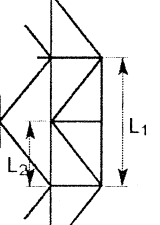
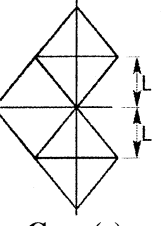
k should be determined taking account of both the bracing pattern (see Figure H.1) and the connections of the bracing to the legs. In the absence of more accurate information values of k should be obtained from table G.2.

(c) *Horizontal bracing members*

In the case of horizontal members of K bracing without plan bracing (see H.3.10) that have compression in one half of their length and tension in the other, the effective slenderness factor k for buckling

transverse to the frame determined from table G.2, should be multiplied by the factor k_1 given in table G.3 depending on the ratio of the tension load, N_t , to the compression load N_c .

Table G.1 Effective slenderness factor k for leg members

Symmetrical bracing			Unsymmetrical bracing			
Section	 ⁽³⁾		Section	 ⁽³⁾		
Axis	v - v	y - y	Axis	v - v	y - y	y - y
 Case (a) Primary bracing at both ends	$0,8 + \frac{\bar{\lambda}}{10}$ but $\geq 0,9$ and $\leq 1,0$	$1,0^{(1)}$	 discontinuous top end with horizontals  Case (d) Primary bracing at both ends	$1,2 \left(0,8 + \frac{\bar{\lambda}}{10} \right)$ but $\geq 1,08$ and $\leq 1,2$ on $L_2^{(2)}$	$1,2 \left(0,8 + \frac{\bar{\lambda}}{10} \right)$ but $\geq 1,08$ and $\leq 1,2$ on L_1	$1,0$ on $L_1^{(1)}$
 asymmetric  symmetric Case (b) Primary bracing at one end and secondary bracing at the other	$0,8 + \frac{\bar{\lambda}}{10}$ but $\geq 0,9$ and $\leq 1,0$	$1,0^{(1)}$	  Case (e) Primary bracing at both ends	$0,8 + \frac{\bar{\lambda}}{10}$ but $\geq 0,9$ and $\leq 1,0$ on $L_2^{(2)}$	$0,8 + \frac{\bar{\lambda}}{10}$ but $\geq 0,9$ and $\leq 1,0$ on L_1	$1,0$ on $L_1^{(1)}$
 Case (c) Secondary bending at both ends	$0,8 + \frac{\bar{\lambda}}{10}$ but $\geq 0,9$ and $\leq 1,0$	$1,0$				

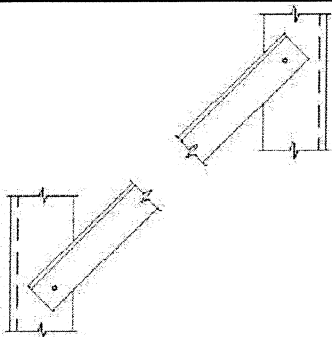
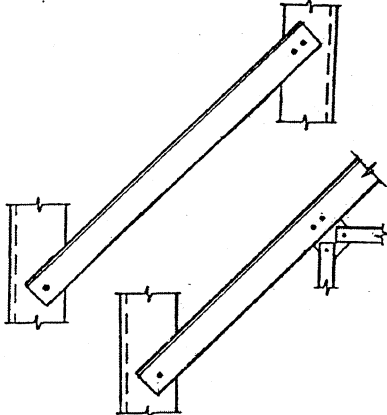
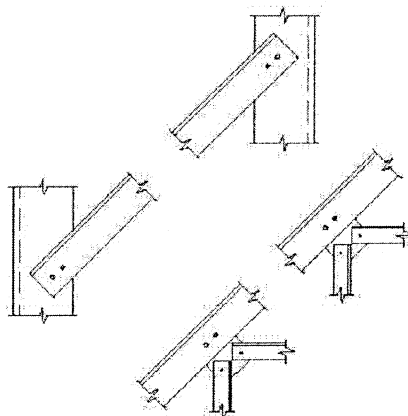
NOTE 1: A reduction factor may be justified by analysis.

NOTE 2: Only critical if very unequal angle section is used.

NOTE 3: The above values only apply to 90° angles.

Table G.2 Effective slenderness factor k for bracing members

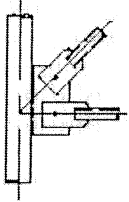
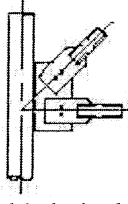
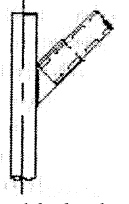
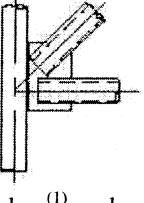
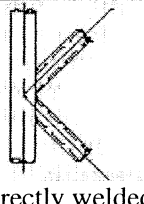
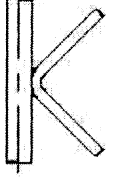
(a) Single and double bolted angles

Type of restraint	Examples	Axis	k
Discontinuous both end (i.e. single bolted at both ends of member)		v-v	$0,7 + \frac{0,35}{\lambda_v}$
		y-y	$0,7 + \frac{0,58}{\lambda_y}$
		z-z	$0,7 + \frac{0,58}{\lambda_z}$
Continuous one end (i.e. single bolted at one end and either double bolted or continuous at other end of member)		v-v	$0,7 + \frac{0,35}{\lambda_v}$
		y-y	$0,7 + \frac{0,40}{\lambda_y}$
		z-z	$0,7 + \frac{0,40}{\lambda_z}$
Continuous both ends (i.e. double bolted at both ends, double bolted at one end and continuous at other end, or continuous at both ends of the member)		v-v	$0,7 + \frac{0,35}{\lambda_v}$
		y-y	$0,7 + \frac{0,40}{\lambda_y}$
		z-z	$0,7 + \frac{0,40}{\lambda_z}$

NOTE 1: Above details are shown for illustrative purposes only and may not reflect practical design aspects.

NOTE 2: Details are shown for connections to angle legs. The factor K applies equally to connections to tubular or solid round legs through welded gusset plates.

Table G.2 Effective slenderness factor k for bracing members
(b) Tubes and rods

	Type	Axis	$K^{(3)(5)}$
tube or rod legs	 single bolted tube	in plane	0,95 ⁽²⁾
		out of plane	0,95 ⁽²⁾
	 double bolted tube	in plane	0,85
		out of plane	0,95 ⁽²⁾
	 welded tubes with end plates	in plane	0,85
		out of plane	0,95 ⁽²⁾
	 welded tubes ⁽¹⁾ and rods with welded gussets	in plane	0,70
		out of plane	0,85
	 directly welded tubes and rods	in plane	0,70
		out of plane	0,70
	 welded bent rods	in plane	0,85
		out of plane	0,85

NOTE 1: Double preloaded bolts may qualify for this condition subject to analysis.

NOTE 2: Reduction for actual length only, but not less than the distance between end bolts.

NOTE 3: Where ends are not the same, an average " K " value should be used.

NOTE 4: Above details are shown for illustrative purposes only and may not reflect practical design aspects.

NOTE 5: Above values are for bracing members with the same connection type at each end. For members with intermediate secondary bracing " K " factors may increase and upper values of 1,0 should be used unless justified by tests.

Table G.3 Modification factor (k_1) for horizontal of K brace without plan bracing

Ratio $\frac{N_t}{N_c}$	Modification factor, k_1
0,0	0,73
0,2	0,67
0,4	0,62
0,6	0,57
0,8	0,53
1,0	0,50
A value of 1,0 applies when the ratio $\frac{N_t}{N_c}$ is negative (i.e. when both members are in compression).	

Annex H [informative] – Buckling length and slenderness of members

H.1 General

(1) This annex gives information about the determination of buckling length and slenderness of members in masts and towers.

H.2 Leg members

- (1) The slenderness for leg members should generally be not more than 120.
- (2) For single angles, tubular sections or solid rounds used for leg sections with axial compression braced symmetrically in two normal planes, or planes 60° apart in the case of triangular structures, the slenderness should be determined from the system length between nodes.
- (3) Where bracing is staggered in two normal planes or planes 60° apart in the case of triangular structures, the system length should be taken as the length between nodes. The slenderness for the case shown in table G.1, case (d) should be determined from equation (H.1a) or (H.1b) as appropriate. The slenderness should be taken as:

$$\lambda = \frac{L_1}{i_{yy}} \text{ or } \lambda = \frac{L_2}{i_{vv}} \text{ for angles} \quad \dots \text{ (H.1a)}$$

$$\lambda = \frac{L_1}{i_{yy}} \text{ for tubes} \quad \dots \text{ (H.1b)}$$

NOTE: The value $\lambda = \frac{L_2}{i_{vv}}$ may be conservative in relation to a more refined analysis taking account of realistic end conditions.

- (4) Built-up members for legs may be formed with two angles in cruciform section or of two angles back to back.
- (5) Built-up members consisting of two angles back to back (forming a T) may be separated by a small distance and connected at intervals by spacers and stitch bolts. They should be checked for buckling about both rectangular axes according to 6.4.4 of EN 1993-1-1. For the maximum spacing of stitch bolts, see EN 1993-1-1, 6.4.4.

NOTE: The National Annex may give information on procedures where the maximum spacing of the stitch bolts is larger than that given in EN 1993-1-1, 6.4.4.

- (6) Stitch bolts should not be assumed to provide full composite action where the gap between the angles exceeds 1,5 t , and the properties should be calculated assuming a gap equal to the true figure or 1,5 t , whichever is the lesser where t is the thickness of the angle. If batten plates are used in addition to stitch bolts the properties corresponding to the full gap should be taken. See 6.4.4 of EN 1993-1-1.
- (7) Battens should prevent relative sliding of the two angles; if bolted connections of categories A and B are used, see 3.4 of EN 1993-1-8, the bolt hole diameter should be reduced.

NOTE 1: The rules (5) to (7) also apply to built-up members in bracings.

NOTE 2: The National Annex may give further information.

H.3 Bracing members

H.3.1 General

(1) The following rules should be used for the typical primary bracing patterns shown in Figure H.1. Secondary bracings may be used to subdivide the primary bracing or main leg members as shown, for example, in Figures H.1 (IA, IIA, IIIA, IVA) and H.2.

(2) The slenderness λ for bracing members should be taken as:

$$\lambda = \frac{L_{di}}{i_{vv}} \text{ for angles} \quad \dots \text{ (H.2a)}$$

$$\lambda = \frac{L_{di}}{i_{yy}} \text{ for tubes} \quad \dots \text{ (H.2b)}$$

where L_{di} is specified in Figure H.1

NOTE: The value $\lambda = \frac{L_{di}}{i_{vv}}$ may be conservative in relation to a more refined analysis taking account of realistic end conditions.

(3) The slenderness λ for primary bracing members should generally be not more than 180 and for secondary bracing not more than 250. For multiple lattice bracing (Figure H.1(V)) the overall slenderness should generally be not more than 350.

NOTE: The use of high slenderness ratios can lead to the possibility of individual members vibrating and can make them vulnerable to damage due to bending from local loads.

H.3.2 Single lattice

(1) A single lattice may be used where the loads are light and the lengths relatively short, as for instance near the top of towers or in light masts (see Figure H.1(I)).

Typical primary spacing patterns					
parallel or tapering			usually tapering		usually parallel
I	II	III	IV	V	VI
Single lattice	Cross bracing	K-bracing	Discontinuous bracing with continuous horizontal intersections	Multiple lattice bracing	Tension bracing
$L_{di} = L_d$	$L_{di} = L_{d2}$	$L_{di} = L_{d2}$	$L_{di} = L_{d2}$		
Typical secondary bracing patterns (see also Figure H.2)				NOTE: The tension members in pattern VI are designed to carry the total shear in tension, e.g. <div style="text-align: center;"> or </div>	
IA	IIA	IIIA	IVA		
	Cross bracing	K-bracing	Cross bracing with secondary members		
	$L_{di} = L_{d1}$	$L_{di} = L_{d1}$ $L_{di} = L_{d2}$ on rectangular axis	$L_{di} = L_{d1}$		

Figure H.1 Typical bracing patterns

H.3.3 Cross bracing

(1) Provided that the load is equally split into tension and compression, the members are connected where they cross, and provided also that both members are continuous (see Figure H.1(II)), the centre of the cross may be considered as a point of restraint both transverse to and in the plane of the bracing and the critical system length becomes L_{d2} on the minor axis.

(2) Where the load is not equally split into tension and compression and provided that both members are continuous, the compression members should be checked in the same way for the largest compressive force. In addition, it should be checked that the sum of the buckling resistances of both members in compression is at least equal to the algebraic sum of the axial forces in the two members. For the calculation of the buckling

resistances, the system length should be taken as L_d and the radius of gyration as that about the rectangular axis parallel to the plane of the bracing. The slenderness may be taken as:

$$\lambda = \frac{L_d}{i_{yy}} \text{ or } \frac{L_d}{i_{zz}} \quad \text{for angles} \quad \dots \text{ (H.3a)}$$

$$\lambda = \frac{L_d}{i_{yy}} \quad \text{for tubes or solid rounds} \quad \dots \text{ (H.3b)}$$

NOTE: Where either member is not continuous, the centre of the connection may only be considered as a restraint in the transverse direction if the detailing of the centre connection is such that the effective lateral stiffness of both members is maintained through the connection and the longitudinal axial stiffness is similar in both members.

H.3.4 Tension bracing

- (1) Each diagonal member of a pair of tension bracing members and the horizontals should be capable of carrying the full bracing shear load (see Figure H.1 (VI)).

NOTE: Tension systems are very sensitive to methods of erection and to modifications or relative movements. Detailing to give an initial tension within the bracing and to provide mutual support at the central cross will be required to minimise deflection.

H.3.5 Cross bracing with secondary members

- (1) Where secondary members are inserted to stabilize the legs (see Figure H.1(IIA and IVA) and Figure H.2(a)), the buckling length on the minimum axis should be taken as L_{d1} .
- (2) Buckling should also be checked over length L_{d2} on the rectangular axis for buckling transverse to the bracing and then over length L_d for the algebraic sum of the axial forces, see H.3.3.

H.3.6 Discontinuous cross bracing with continuous horizontal at centre intersection

- (1) The horizontal member should be sufficiently stiff in the transverse direction to provide restraints for the load cases where the compression in one member exceeds the tension in the other or where both members are in compression, see Figure H.1 (IV).
- (2) This criterion may be satisfied by ensuring that the horizontal member withstands (as a compression member over its full length on the rectangular axis) the algebraic sum of the axial force in the two members of the cross-brace, resolved in the horizontal direction.

NOTE: Additional allowance can be necessary for the bending stresses induced in the edge members by local loads transverse to the frame, such as wind.

H.3.7 Cross bracing with diagonal corner members

- (1) In some patterns of cross bracing a corner member may be inserted to reduce the buckling length transverse to the plane of bracing (see Figure H.2(b)). A similar procedure to that used for H.3.3 may be used to determine whether this will provide a satisfactory restraint.
- (2) In this case five buckling checks should be carried out as follows:
 - Buckling of member against the maximum load over length L_{d1} on the minimum axis;
 - Buckling of member against the maximum load over length L_{d2} on the transverse rectangular axis;
 - Buckling of two members in cross brace against the algebraic sum of loads in cross brace over the length L_{d3} on the transverse axis;
 - Buckling of two members (one in each of two adjacent faces) against the algebraic sum of the loads in the two members connected by the diagonal brace over length L_{d4} on the transverse axis.

NOTE: For this case the total resistance should be calculated as the sum of the buckling resistances of both members in compression (see H.3.3(2)).

- Buckling of four members (each member of cross brace in two adjacent faces) against the algebraic sum of loads in all four members over length L_d on the transverse axis.

H.3.8 Diagonal members of K bracing

(1) In the absence of any secondary members (see Figure H.1(III)) the critical system length may be taken as L_{d2} on the minor axis.

(2) Where secondary bracing in the faces is provided but no hip bracing (see Figure H.1(IIIA)) the critical system length should be taken as L_{d2} on the appropriate rectangular axis. Thus the slenderness should be taken as:

$$\lambda = \frac{L_{d2}}{i_{yy}} \text{ or } \frac{L_{d2}}{i_{zz}} \quad \dots \text{ (H.4)}$$

(3) Where secondary bracing and triangulated hip bracing is provided (see Figure H.2(c)), then the appropriate system length between such hip members L_{d4} should be used for checking buckling transverse to the face bracing on the appropriate rectangular axis. Thus the slenderness may be taken as:

$$\lambda = \frac{L_{d4}}{i_{yy}} \text{ or } \frac{L_{d4}}{i_{zz}} \text{ for all types of section} \quad \dots \text{ (H.5)}$$

H.3.9 Horizontal face members with horizontal plan bracing

(1) Where the length of the horizontal face members becomes large, plan bracing may be introduced to provide transverse stability.

(2) The system length of the horizontal member for buckling should be taken as the distance between intersection points in the plan bracing for buckling transverse to the frame, and the distance between supports in plan for buckling in the plane of the frame.

(3) Care should be taken in the choice of the vv or rectangular axes for single angle members. The vv axis should be used unless suitable restraint by bracing is provided at or about the mid-point of the system length. In this case buckling should be checked about the vv axis over the intermediate length and about the appropriate rectangular axis over the full length between restraints on that axis.

NOTE: This procedure may be conservative in relation to a more refined analysis taking account of realistic end conditions.

(4) Where the plan bracing is not fully triangulated, additional allowance should be made for the bending stresses induced in the edge members by loads, such as wind transverse to the frame, see Figure H.3.

(5) To avoid buckling, where the plan bracing is not fully triangulated:

- the horizontal plan bracing should be designed to resist a concentrated horizontal force of $p \times H$ applied at the middle of the member where p is the percentage of the maximum axial compression force, H , in the members of the horizontal plan bracing (see H.4);
- the deflection of horizontal plan bracing under this force should not exceed $L/500$.

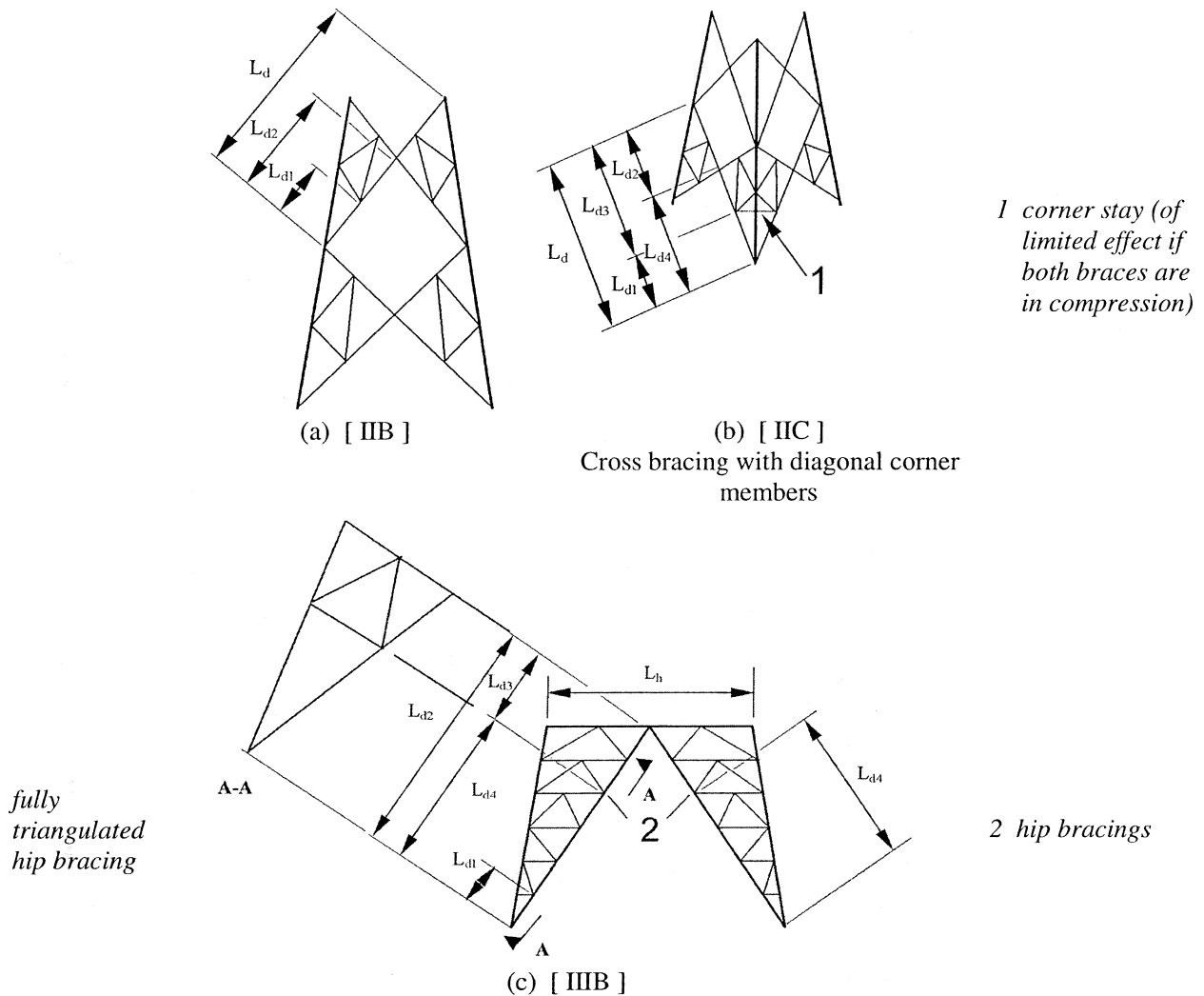


Figure H.2 Use of secondary bracing systems

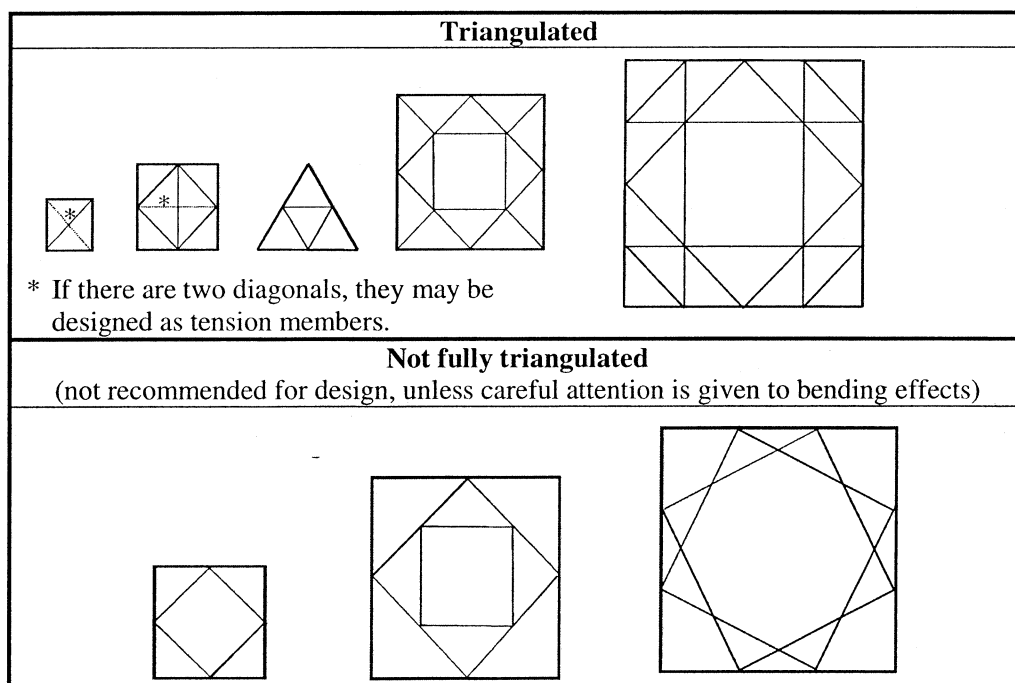


Figure H.3 Typical plan bracing

H.3.10 Horizontal members without plan bracing

(1) For small widths of towers and for masts plan bracing may be omitted in appropriate cases with due justification.

(2) The rectangular radius of gyration should be used for buckling transverse to the frame over length L_h (see Figure H.4(a)). However for single angle members, the radius of gyration about the vv axis should be used over length L_{h2} unless restraint by secondary bracing at intervals along the length is provided in which case the system length should be taken as L_{h1} , see Figure H.4(b).

NOTE: This procedure may be conservative in relation to a more refined analysis taking account of realistic end conditions.

(3) To avoid buckling of the horizontal member the criteria of H.3.9(5) should be satisfied.

NOTE: Additional allowance may be necessary for the bending stresses induced in the edge members by local loads transverse to the frame, such as wind.

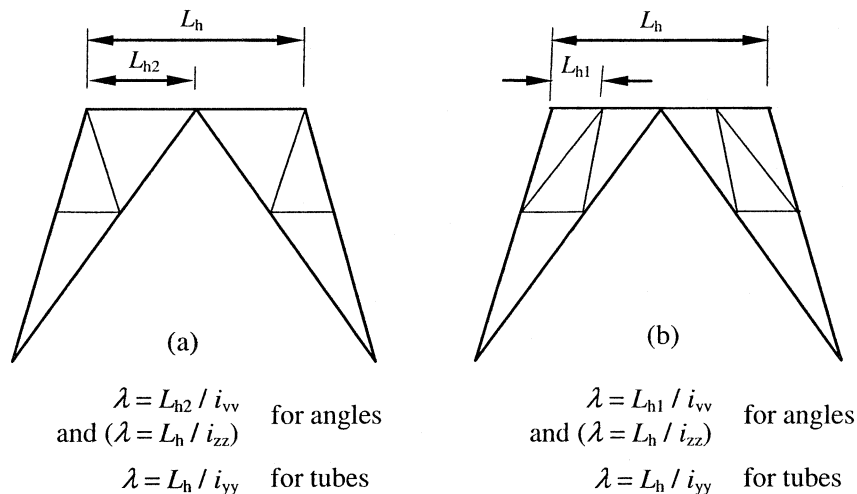


Figure H.4 K bracing horizontals without plan bracing

H.3.11 Cranked K bracing

(1) For large tower widths, a crank or bend may be introduced into the main diagonals (see Figure H.5), which has the effect of reducing the length and size of the redundant members. As this produces high stresses in the members meeting at the bend, transverse support should be provided at the joint. Diagonals and horizontals should be designed as for K bracing, system lengths of diagonals being related to the lengths to the knee joint.

H.3.12 Portal frame

(1) A horizontal member may be introduced at the bend to turn the panel into a portal frame, see Figure H.6. Because this leads to a lack of articulation in the K brace, special consideration should be given to the effects of foundation settlement or movement.

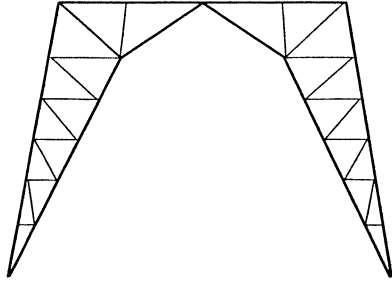


Figure H.5 Cranked K bracing

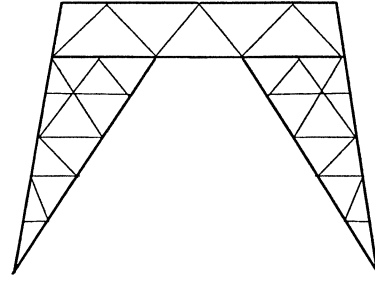


Figure H.6 Portal frame

H.3.13 Multiple lattice bracing

(1) In a multiple lattice configuration the bracing members that are continuous and connected at all intersections should be designed as secondary members (see H.4) on a system length from leg to leg with the appropriate radius of gyration i_{yy} or i_{zz} , see Figure H.7. For the stability of the panel the overall slenderness

$\frac{L}{i_{yy}}$ should be less than 350. For single angle members $\frac{i_{yy}}{i_{vv}}$ should be greater than 1,50 where i_{yy} is the radius of gyration about the axis parallel to the plan of the lattice.

(2) The stability of the member A-B shown in Figure H.7 should be checked under the applied force on the critical system length L_o for the slenderness:

$$\lambda = \frac{L_o}{i_{vv}} \text{ for angles} \quad \dots \text{ (H.6a)}$$

$$\lambda = \frac{L_o}{i_{yy}} \text{ for tubes and solid rounds} \quad \dots \text{ (H.6b)}$$

NOTE: The value of $\lambda = \frac{L_o}{i_{vv}}$ may be conservative in relation to a more refined analysis taking account of realistic end conditions.

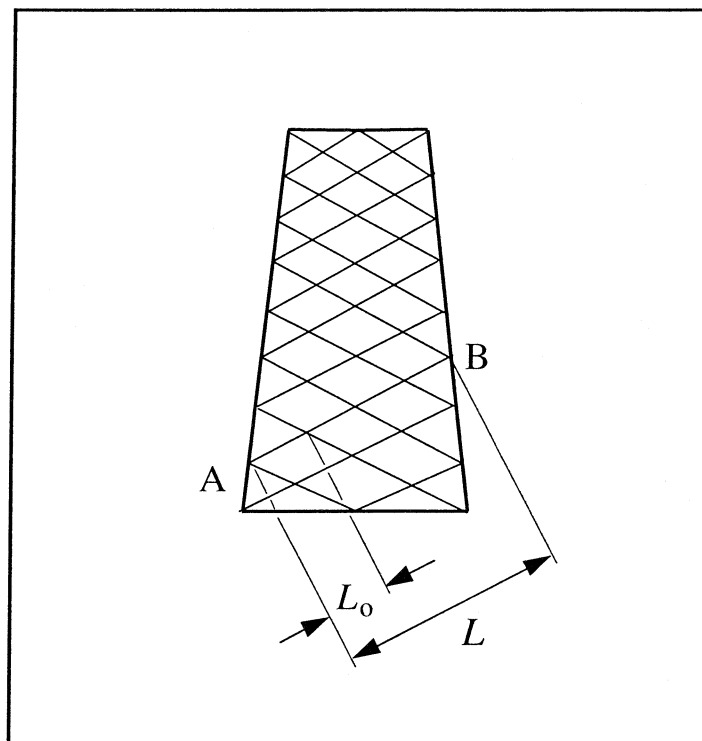


Figure H.7 Multiple lattice bracing

H.4 Secondary bracing members

(1) In order to allow for imperfections in leg members, and for the design of secondary bracing members, a notional force should be introduced acting transverse to the leg member (or other chord if not a leg) being stabilized at the node point of the attachment of the bracing member. Depending on the slenderness of the leg member being stabilized, the value of the notional force to be used for the design of any secondary member should be obtained from (2) and (3).

(2) The force to be applied at each node in turn in the plane of bracing, expressed as a percentage, p , of the axial force in the leg for various values of the slenderness λ of the leg may be taken as:

$$p = 1,41 \text{ when } \lambda < 30 \quad \dots \text{ (H.7a)}$$

$$p = \frac{(40 + \lambda)}{50} \text{ when } 30 \leq \lambda \leq 135 \quad \dots \text{ (H.7b)}$$

$$p = 3,5 \text{ when } \lambda > 135 \quad \dots \text{ (H.7c)}$$

(3) When there is more than one intermediate node in a panel then the secondary bracing system should be checked separately for 2,5% of the axial force in the leg shared equally between all the intermediate node points. These notional forces should be assumed to act together and in the same direction, at right angles to the leg and in the plane of the bracing system.

(4) In both cases (2) and (3) the distribution of forces within the triangulated secondary bracing panel should be determined by linear elastic analysis.

(5) The effects of this notional force should generally be added to the primary force as calculated from the global analysis for the design of any primary member. Exceptionally for self-supporting lattice towers of conventional configuration the notional forces need not be added to the primary forces, provided that the

primary bracing is checked for the effects of the notional force, when the primary force is smaller. For guyed masts the effects of the notional force should always be added to the primary force.

(6) Provided that it is designed for notional forces as described in (1) to (5) it may be assumed that the stiffness of the bracing system will be sufficient.

(7) If the main member is eccentrically loaded or the angle between the main diagonal of a K brace and the leg is less than 25° then the above value of the notional force may be insufficient and a more refined value should be obtained by taking into account the eccentricity moment and secondary stresses arising from leg deformation.

(8) Where the direction of buckling is not in the plane of the bracing, then the values given by equations H7 a), b) and c) should be divided by a factor of $\sqrt{2}$.

H.5 Shell structures

(1) For the strength and stability of shell structures see EN 1993-1-6.

NOTE: See also EN 1993-3-2.

